

3차원 모델빌더와 실시간 시뮬레이션 환경 개발

3D Model Builder and Realtime Simulation Environment Development

*이 지 우, 이 창 훈, 이 덕 재, **이 명 수

Lee Ji-woo, Lee Chang-hoon, Lee Duck-jae, Lee Myeong-soo

Abstract - 국내 및 국외에서 많은 시뮬레이션 환경이 개발되어 다양한 시뮬레이터 개발에 활용되고 있다. 이러한 시뮬레이션 환경은 다양한 기능을 갖추고 있어 시뮬레이터 개발편의를 도모하고 있다. 이러한 시뮬레이션 환경에서는 시뮬레이션 스케줄링(Scheduling), 데이터베이스(Database), 모델 컴파일 그리고 디버거(모니터링 및 제어기능 툴) 등을 제공한다. 이미 국내 및 국외 업체에서 시뮬레이션 환경이 개발되어 왔지만, 3차원적인 시뮬레이션을 필요로 하는 시뮬레이션 환경은 미비한 상태로 로봇공학이나 3차원 시뮬레이션이 필요한 산업에 앞으로 더욱 더 개발이 필요한 실정이다. 본 논문에서는 3차원 시뮬레이션을 하기 위한 방안으로 개발된 3D 모델빌더와 시뮬레이션 환경인 3DSIM(3D Simulation Environment)을 소개한다. 3DSIM은 현재 사용되고 있는 시뮬레이션 환경과는 다른 형식의 구조를 갖추고 있어 일반 시뮬레이션 환경과의 차이점에 대해 알아본다. 일반적인 시뮬레이션 환경은 서버와 클라이언트로 나뉘어 서버에서는 특정 모델에 대한 연산을 수행하기 위한 환경을 제공하며, 클라이언트에서는 서버에서 수행된 연산을 효율적으로 분석하기 쉽도록 보여주는 환경으로 나뉜다. 하지만, 3차원 시뮬레이션 환경인 3DSIM은 3차원 객체를 디자인 하는 과정이 필요하며, 3차원 객체를 디자인 하였을 경우, 각각의 파라미터를 통하여 시뮬레이션 되어야 한다는 점에서 일반적인 시뮬레이션 환경과의 구조가 다르게 된다. 3DSIM개발을 통하여 3차원 시뮬레이터 개발을 보다 편하게 개발할 수 있도록 하였으며, 시뮬레이션 모니터링 및 제어가 용이하도록 개발하였다.

Key Words : 3D Model Builder, Simulation Environment, Robotics

1. 서 론

1.1 3D 모델빌더 및 시뮬레이션 환경 개요

기존의 시뮬레이션 환경은 서버(Server)측에서 모델을 개발하고 클라이언트(Client)측에서는 모델의 시뮬레이션 결과를 GUI(Graphic User Interface)로 표현하는 방식으로 이루어져 개발되어 왔으며, 최근에는 서버환경에 GUI(Graphic User Interface)환경이 포함되어 개발되고 있다. 시뮬레이션 환경에 있어 타이머 스케줄(Schedule)은 필수적 요소이며, 이 외에도 제공해 주어야 할 기능으로는 데이터베이스 관리(Database Management)와 시뮬레이션 상태 제어 등이 있을 수 있다. 기존 시뮬레이션 환경으로는 GSE사의 SSP3.3과 DS&S사의 Simport등이 있다. 이러한 시뮬레이션 환경은 3차원적인 시뮬레이션 모델을 표현하기에는 적합하지 않다. 본 논문에서 소개하는 3DSIM(3D Simulation Environment)은 3차원적인 시뮬레이션에 목적을 두어 기존의 시뮬레이션 환경에서 제공되던 기능을 추가함으로서 효율적인 3차원 모델개발 환경을 갖추도록 하였다.

3DSIM(3D Simulation Environment)은 Microsoft Windows 운영체제를 바탕으로 3차원 라이브러리인 OpenGL을 사용하였으며, MFC(Microsoft Fundamental Class)를 활용하여 개발하였다. 본 연구의 개발 결과물인 3차원 모델빌더와 실시간 시뮬레이션 환경은 시뮬레이션 상태의 IC (Initial

Condition) 상태를 저장하거나 불러오고, 시뮬레이션 중 이전의 상태로 다시 돌아가도록 하는 (Backtrack)기능, 시뮬레이션의 실행(Run), 중지(Freeze), 특정 스텝(Step) 단위의 연산이 가능하며, 사용자에게 쉽게 시뮬레이션하고 접근 할 수 있도록 하는 환경을 개발하였다. 또한 정해진 스케줄에 따라 시뮬레이션되어 정밀한 시뮬레이션이 가능하도록 하였다. 본 시뮬레이션 환경(3DSIM)을 통하여 특정 모델을 생성 후 시뮬레이션 및 제어함으로서 3DSIM을 검증을 하였다.

2. 3D 모델빌더 및 시뮬레이션 환경 개발

2.1 환경 개요

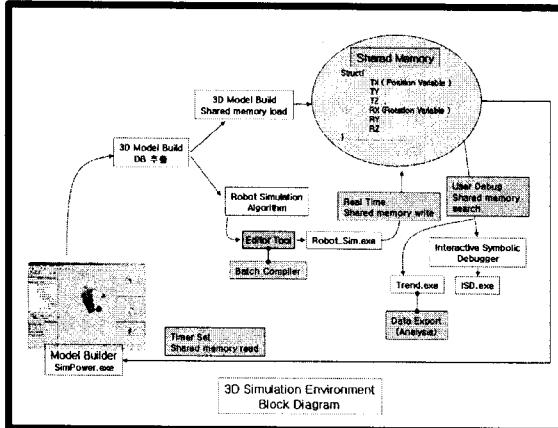
시뮬레이션이 갖추어야 할 기본적인 조건은 경우에 따라 달라 질수 있으나, 본 연구에서 개발된 시뮬레이션 환경인 3DSIM은 아래와 같은 특징을 가진다.

- ◆ 데이터베이스(Database) 관리
 - 시뮬레이션 중에 모든 변수는 공유 메모리에 상주.
- ◆ 소프트웨어 단계 (Software Level)
 - 특정 모델에 대한 선정 3D 모델빌더를 이용한 디자인 단계와 모델의 알고리즘 생성 단계로 분리.
- ◆ 시뮬레이션 프로그램 구조
 - 공유 메모리를 이용한 프로그램간의 상호 인터페이스.
- ◆ 디버거 툴 ISD(Interactive Symbolic Debugger) 제공
 - 현재 상태를 모니터링하고 제어하는 명령 제공.
 - 실행(Run), 중지(Freeze), 스텝(Step) 및 그 이외의 명령어 제공.
- ◆ 데이터 획득 및 분석 툴 제공

저자 소개

- * 이 지 우 : 倍材大學校 電子工學科 碩士課程
- 이 창 훈 : 倍材大學校 電子工學科 博士
- 이 덕 재 : 倍材大學校 電子工學科 碩士課程
- ** 이 명 수 : 電力研究院 責任研究員

3차원 모델빌더와 실시간 시뮬레이션 환경인 3DSIM의 전반적인 구성을 간략하게 설명하면 아래의 블록 다이어그램(Block Diagram)을 보면 알 수 있듯이 모델빌더(Model Builder) 하나에 여러 개의 프로그램이 함께 공유 메모리를 이용하여 동작함을 알 수 있다.



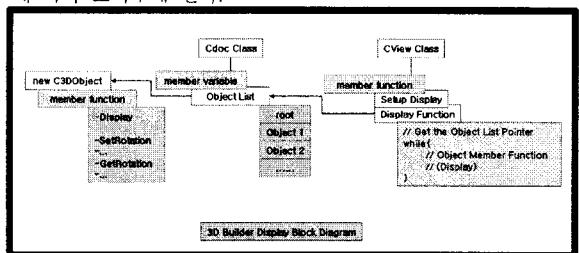
[그림 1] 3차원 모델빌더와 실시간 시뮬레이션 환경 구조

위의 블록 다이어그램(Block Diagram)에서 보는 것과 같이 모델빌더(Model Builder)인 3D Builder는 3차원 물체를 모델링 할 수 있는 기능을 가지고 있으며, 부모와 자식간의 링크연결을 통하여 디자인 된다. 또한 각각의 원소는 데이터베이스 관리(Database Management)기능을 가지고 있어 자동으로 각원소의 이동 및 회전변수가 추출된다. 추출된 DB는 사용자가 알고리즘 구현시 사용되며, 사용자가 보다 쉽게 알고리즘에 접근하기 위해 Precompiler인 제작 툴(Editor Tool)을 제공한다. 제공된 툴을 이용하여 모델을 쉽게 생성 가능하며, 3차원 모델빌더는 생성된 모델(ModelBuild.dll)을 타이머 스캐줄에 따라 제어하게 된다. 3차원 모델빌더와 생성된 모델은 공유메모리를 통하여 데이터가 상호 인터페이스(Interface)되며, 시뮬레이션에 대한 모션을 표현한다. 모델빌더에서 공유메모리를 생성하고, 시뮬레이션 실행시 모델은 연산된 결과를 공유메모리에 기록하고, 기록된 결과를 모델빌더(Model Builder)에서 읽어와 표현(Display)하게 된다.

2.2 모델빌더 및 시뮬레이션 환경 구조 및 기능

모델빌더는 OpenGL 라이브러리와 MFC(Microsoft Fundamental Class)를 사용하여 제작되었으며, 모델빌더의 주요 클래스는 C3DObject, CSimRobotDoc, CSimRobotView이다. Doc & View 클래스의 구조를 갖는 모델빌더는 Doc 클래스에서는 현재 설정된 화면의 조명, 배경색 및 그 외의 데이터 정보를 가지고 있으며, View 클래스에서는 Doc 클래스의 데이터 정보를 참조하여 사용자에게 3차원적으로 보여주는 클래스이다. OpenGL라이브러리의 함수를 사용하여 사용자에게 3차원 모형을 만들어 준다. C3DObject 클래스는 객체의 속성을 정의하는 클래스로 멤버 함수로 SetRotation, SetPosition, GetRotation, GetPosition등이 있어 새로운 객체를 생성 후 객체의 속성을 변경하여 객체의 좌표를 재구성하게 된다. Display()라는 멤버함수는 현재 설정된 객체 속성

에 따라 보여주게 된다.

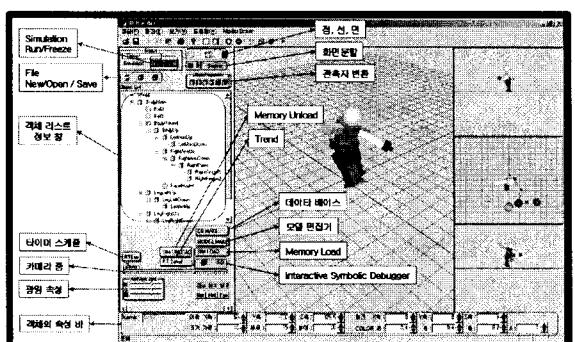


[그림 2] 모델빌더의 클래스 구조

위의 블록 다이어그램에서 보는 것과 같이 CDoc 클래스에서 C3DObject 객체의 리스트를 담당하는 Object List 변수가 있음을 알 수 있다. 사용자로부터 객체의 추가, 변경 및 제거함에 따라 리스트의 정보가 변경된다. CView 클래스에서는 변경된 리스트 포인터를 획득하여 특정 함수에서 순차적으로 객체의 멤버함수인 Display()를 호출하여 객체의 순차적인 디스플레이를 함으로서 새롭게 추가되거나 변경된 내용이 적용된다. 전체적인 모델빌더의 간략화한 디자인에 대한 구조만을 보면 위의 그림과 같으며, 실제 제작된 모델 디자인의 기능을 보면 아래 그림과 같이 구성되어 실행된다.

모델빌더는 객체를 디자인할 수 있도록 제작되었으며, 디자인은 여러 가지 프IMITIVE(Primitive)을 이용하여 디자인 가능하며, 이렇게 생성된 객체를 파일로 저장하고 불러오는 기능이 기본적으로 제공되며, 이 외에도 다양한 기능을 모델빌더가 제공한다. 실제 제작된 모델빌더의 실행 모습은 다음과 같으면 모델빌더가 제공하는 기능을 나열하면 다음과 같다.

- ◆ 상위계층과 하위계층에 대한 속성 설정을 통한 링크 연결.(부모와 자식간의 계층적 구조 생성)
- ◆ 사용자가 모델을 디자인할 수 있는 최소한의 기능.(객체의 생성, 제거, 수정)
- ◆ 모델 디자인 후 파일 저장 및 오픈 기능.
- ◆ 각 모델에 대한 속성을 이용한 데이터베이스 생성.
- ◆ 3차원 화면의 속성변경(광원, 관측자 변환, 분할화면, 다양한 보기(점,선,면))기능.
- ◆ 타이머 스캐줄 속성 변경 기능.
- ◆ 시뮬레이션 실행, 정지 기능.
- ◆ 시뮬레이션 분석 도구 실행 (ISD, RTTrend)기능.



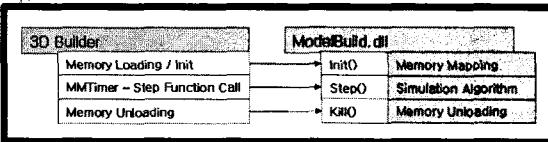
[그림 3] 시뮬레이션 프로그램 실행 모습

가장 기본이 되는 디자인 단계에서 상위계층과 하위계층을 이용한 링크연결을 통하여 모델을 디자인을 할 수 있으며, 객체의 디자인시 필요한 기본적인 기능으로 새로운 객체의 동

적 생성, 생성된 객체의 속성 변경 그리고 삭제 기능을 지원한다. 또한 객체의 복사 및 복이기 기능을 갖고 있다. 이렇게 생성된 동적 객체를 파일로 저장하여 불러올 수 있으며, 새롭게 디자인된 모델은 객체의 속성에 따라 데이터베이스를 추출하며, 추출된 데이터베이스와 사용자가 추가하고자 하는 애널로그 변수와 디지털 변수를 등록하여 모델 빌드시 사용한다. 모델 빌더는 3차원 속성을 변경하여 사용자의 구미에 맞도록 변경하여 사용할 수 있도록 하였다. 타이머 스케줄을 변경하는 기능을 갖고 있으며, 이렇게 변경된 정보를 이용하여 시뮬레이션의 실행 및 정지 명령이 가능하게 된다. 그 외에 시뮬레이션 제어 및 분석용 도구인 ISD(Interactive Symbolic Debugger), RTTrend(Realtime Trend) 프로그램을 실행할 수 있도록 하였다.

2.3 시뮬레이션 모델 생성 및 파일의 구조

시뮬레이션 모델은 본 환경에서 제공하는 에디터 툴을 이용하여 생성 가능하다. 시뮬레이션 모델 빌드를 통하여 생성된 ModelBuild.dll 파일은 모델 빌더에서 로드 및 실행을 담당한다. 시뮬레이션 모델 파일은 Init(), Step(), Kill() 함수로 이루어져 있다. 이 함수들은 모델 빌더의 제어에 의해 이루어지며, 모델 빌더와 생성된 모델의 관계의 구조를 보면 다음과 같다.



[그림 4] 모델 빌더와 시뮬레이션 모델의 관계

모델 빌더에서 공유 메모리 생성 후 ModelBuild.dll 파일의 Init() 함수를 호출하여 로드 및 초기화 작업을 한다. 이때 Init() 함수가 호출됨에 따라 데이터베이스 정보를 이용하여 공유 메모리와 매핑이 이루어진다. 모델 빌더에서 설정한 시뮬레이션 스케줄에 따라 ModelBuild.dll 파일에 있는 Step() 함수를 멀티미디어 타이머(Multi Media Timer)에 의해 호출된다. 앞에서 Step() 함수는 알고리즘 구현부분이며, 시뮬레이션 시 공유 메모리에 연산된 결과가 기록됨을 알았다. 또한 시뮬레이션 종료시 Kill() 함수가 호출이 되며, 메모리 로드 및 초기화 작업으로 생성된 메모리가 모두 제거된다.

2.4 스케줄 관리

타이머 스케줄은 시뮬레이션 주기를 설정하여 한 주기에 시뮬레이션 코드를 한번 실행한다. 1초에 12번의 시뮬레이션 루프를 실행하고자 할 경우에는 타이머 스케줄을 83msec로 설정하여 실행하면 된다. 시뮬레이션을 하기 위해서는 타이머 스케줄이 필수적이며, 모델 빌더에서 제공되는 타이머 스케줄은 1msec까지 가능한 멀티미디어 타이머(Multi Media Timer)를 사용하였다. 그리고 시뮬레이션 코드 함수는 ModelBuild.dll 파일의 Step() 함수에 구현되어 있어, 모델 빌더에서 타이머 스케줄 설정값에 따라 Step() 함수를 호출하게 된다. 이때 Step() 함수의 연산 수행 시간은 모델 빌더에서 설정한 스케줄 타임보다 코지 않아야 한다. 만약 1msec로 설정된 스케줄에 시뮬레이션 알고리즘 연산이 1msec를 넘어 설 경우 오버타임 에러(Overtime Error)로 간주된다.

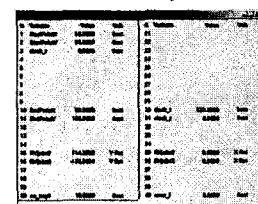
본 시뮬레이션 환경에서는 오버타임 에러(Overtime Error)가 12회 이상 발생될 경우, 현재의 시뮬레이션 결과는 실시간

이 아님을 간주하여 시뮬레이션이 중지된다. 화면 개신의 주기 또한 변환 가능하도록 하였다. 기본적으로 50msec 단위로 설정하였으며, 이는 1초에 20회 화면 개신하게 된다. 3차원 모델의 자연스러운 움직임을 원할 경우 더 작은 값으로 변경하면 되지만, 시뮬레이션 모델 연산이 중요하고, 많은 연산이 필요시에는 화면 개신 횟수를 줄여 모델 연산에 더 가중치를 적용할 수 있도록 하였다.

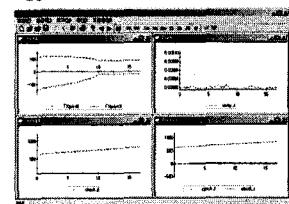
2.5 ISD(Interactive Symbolic Debugger) 및 Trend의 구성

ISD는 현재의 변수 값을 사용자에게 보여주고, 변경할 수 있는 모니터링 프로그램이다. 보여주는 기능뿐만 아니라, 실제 공유 메모리의 데이터를 조작하여, 시뮬레이션의 사고를 가능하게 한다. ISD의 명령어로 시뮬레이션의 실행(run), 정지(freeze), 현재 상태값을 저장하는 스냅(snap), 이전에 저장된 상태값을 불러오는 리셋(reset) 등과 같은 명령어가 있다.

직접 모델 빌더에서 움직이는 3차원 객체를 통하여 시뮬레이션의 결과를 확인할 수 있지만, 정확한 수치 값은 확인하기 어렵다. 이러한 단점을 보완하고 실제 제어하기 위하여 ISD(Interactive Symbolic Debugger) 프로그램을 개발하였다.



[그림 5] ISD 프로그램



[그림 6] 트랜드 실행 모습

ISD는 현재 상태의 수치 값을 확인하고 제어하는 반면, 본 트랜드 프로그램은 이전 값을 누적하여 사용자에게 전체적인 시뮬레이션 과정을 실시간으로 보여줌으로서 시뮬레이션 분석을 보다 용이하게 하였다.

3. 결 론

3차원 모델 빌더와 실시간 시뮬레이션 환경을 통하여 3차원적인 시뮬레이션을 일반 사용자에게도 쉽게 접근할 수 있도록 개발하였다. 특히 3D 빌더를 통한 모델과 시뮬레이션 모델의 모듈화를 통하여 개별적인 시뮬레이션이 가능함으로서 관찰 및 고찰을 용이하도록 하였다. 현재까지 개발된 3DSIM(3D Simulation Environment)은 모델 디자인 기능이 약한 반면, 시뮬레이션의 스케줄링 및 제어 기능의 장점을 갖고 있다. 추후 모델 디자인 기능에 대한 보강과 다양한 3차원 시뮬레이션 모델 적용을 통하여 본 시뮬레이션 환경의 검증이 요구된다.

본 논문은 산업자원부 전력산업 연구개발사업인 “디지털기반 주체어설 인간공학 검증시스템 개발”과제의 연구비 지원에 의하여 연구되었음.

참 고 문 헌

- [1] GSE, SimSuitePower3.3 Training Reference Manual
- [2] DS&S, Simport Training Reference Manual
- [3] (Beer, Ferdinand P.), (E, Russel Johnston, Jr) "Vector Mechanics for Engineers" McGraw-Hill, 1988