Dynamics Simulation Using OpenGL

2000009450 김익준 ・담당교수 : 김재정 교수

Abstract

최근 동역학 시스템의 해석을 하기 위해서 상용 소프트웨어를 사용하는 경우가 종종 있다. 이와 같은 상용 소프트웨어는 상당히 신뢰성 있는 결과를 보여주지만 가격이 고가이고 사용자에게 내부 시뮬레이션 소스가 공개 되있지 않기 때문에 출력된 결과를 그대로 사용자가 받아드리기 힘든 경우도 있다. 이와 같은 단점을 보완하기 위하여 이 논문에서는 OpenGL 을 이용한 프로그램을 가지고 간단하게 동역학 시뮬레이션을 하는 법을 소개한다.

1. Introduction

OpenGL Simulator의 장점은 다음과 같다.

첫째, 수치적인 데이터가 아닌 비주얼한 시뮬레이션 가능하다는 것이다. 기존의 Console 기반의 프로그램이 아니고 MFC Based에서 OpenGL을 이용하여 시뮬레이션 하므로 좋은 품질의 그래픽을 보여줄수 있다. 둘째, 값비싼 소프트웨어가 없이도 간단히 시뮬레이션 가능하다. Recurdyn, Adams 와 같은 고가의 소프트웨어가 없어도 시뮬레이션이 가능하다. 셋째, 시뮬레이션 방법을 사용자가 알 수 있고 변경 가능하다. 사용자가 직접 적분방식이나 운동방정식의 좌표를 설정할 수 있기 때문에 결과를 사용자가 더욱 잘 이해 할 수 있다.

이 프로그램에 적용한 동역학 모델은 Crank Slider, Inverted Pendulum 그리고 회전하는 팽이 모델 이렇게 3가지를 적용하였다. 운동방정식은 Crank Slider를 제외한 나머지 모델에 모두 Lagrangian을 이용하여 풀었으며 적분 방식은 Euler Method 를 사용하였다. 3가지 모델 모두 다 Catia를 이용하여 설계한뒤 STL파일로 저장하여 그 파일의 Vertex 정보를 사용하였다. 이때 예전의 STL2CPP.exe 라는 프로그램을 사용하지 않고 직접 실행중에 Vertex 정보를 추출해 오는 방식을 선택하였다. 이 모든 것이 MFC Single Document Base에서 OpenGL과 직접 설계한 CMatrix Class 와 같이 실행되는 것이다.

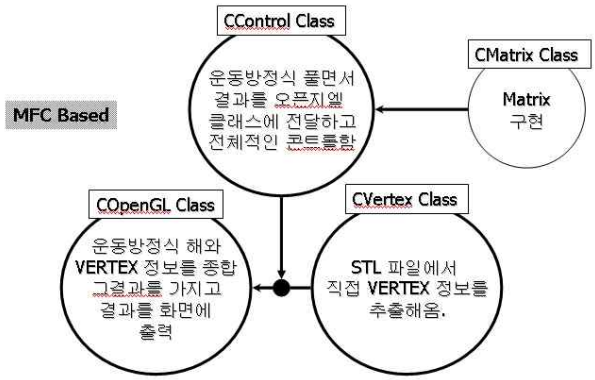
이 논문에서는 운동방정식보다는 프로그램의 작동원리 및 알고리즘에 중점을 두어 설명하겠다.

2. Equations of Motion

운동방정식의 유도는 Lagrangian 을 이용하여 구하였다. 3개 모델에는 각 링크마다 임의의 Damping 계수를 줘서 실제 시스템과 비슷 하도록 설정해 주었다. 이때 Damping 에 관련된 힘을 Lagrangian 에 적용시키기 위하여 Rayleigh Dissapation 함수를 사용하였다. Crank Slider는 가상일의 원리와 라그랑지 승수 를 이용하여 구하였다. (Mechatronics 연구실에서 도움을 많이 받았다.)

3. Class 구조

OpenGL Simulator의 기본 구조는 다음과 같다.



위 그림에서 주목할 만한 부분이 바로 CVertex Class 이다. 이 클래스는 STL 파일에서 직접 Vertex 정보를 가져오는 역할을 하는데 이로 인해서 STL2CPP.exe 보다 훨씬 컴파일 시간을 단축시킬 수 있었다. STL2CPP.exe 같은 경우는 수천줄에 달하는 Vertex 정보를 직접 소스코드에 써 넣어야 하기 때문에 컴파일 시간이 조금만 복잡한 형상일 경우에도 몇 분을 넘어가는 일이 많았다. 하지만 이 프로그램에서는 CVertex Class 를 이용하여 프로그램 실행중에 직접 추출해 오

는 형식이므로 컴파일 하는데 몇초 뿐이 걸리지 않는다. 다음은 CVertex Class의 주요 코드이다.

```
double* CVertex::GetVertexFromSTL(char *filename)
{
      FILE *fp;
      int n,i=0,j=0;
      char ch,str[20];
      double *Temp;
      n=CountData(filename);
      Temp= new double [n];
      fp = fopen(filename,"rt");
      if(fp==NULL) {
            AfxMessageBox("File not Found");
            exit(0);   }

     while(GotoNum(fp)!=NULL)   {
      while((ch=fgetc(fp))!='₩n'&&ch!=NULL){
      if(ch==' ') {
      Temp[i]=atof(str);  i++;
      for(int s=0;s<20;s++)  str[s] = NULL; j=0;

      if(GotoNum(fp)==0)   break;
      }
      else {
                  str[j]=ch;
                  j++;
            }
      }
      Temp[i]=atof(str);  j=0; i++;
      for(int s=0;s<20;s++)  str[s] = NULL;
      } fclose(fp);
      return Temp;
}

int CVertex::CountData(char *filename)
{
      FILE *fp;
      fp = fopen(filename,"rt");
      int ch,i=0;
      if(fp==NULL)
      {
            AfxMessageBox("File not Found");
            exit(0);
      }
      while(GotoNum(fp)!=NULL)
      {
            while((ch=fgetc(fp))!='₩n'&&ch!=NULL)
            {
                  if(ch==' ')
                  {
                        i++;
                        if(GotoNum(fp)==0)   break;
                  } else ;
            } i++;
      }     fclose(fp);       return i;
}
```

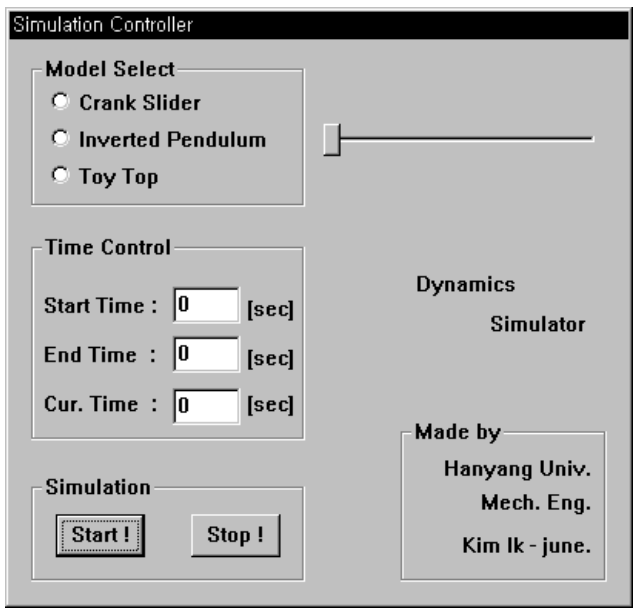
이 Source Code는 STL2CPP.exe 의 내용을 참고 하여 만들었다. 다음으로 소개 할 클래스는 CMatrix 클래스 이다. C++를 이용하여 프로그래밍을 할 때 가장 성가신 부분이 바로 매트릭스 선언과 연산이다. 매트릭스를 선언하기 위해서는 2차원 배열을 이용하여야 하고 매번 동적으로 배열의 공간을 할당 해줘야 하기 때문에 힘이 많이 든다. 또 연산을 위해서는 더하고 빼고 곱하는 모든 연산을 함수로 지정해 놓고 사용해야 하기 때문에 직관적인 프로그래밍이 되지 않는다. 그러한 이유로 매트릭스 클래스를 만들게 되었다. 즉 매트릭스라는 하나의 변수형을 만든 것이다. 이와 같이 하면 다음과 같이 간단하게 매트릭스를 연산하는 프로그램을 코딩할수 있다.

```
CMatrix A(3,3);   // 3X3 매트릭스 선언
CMatrix B(3,1);   // 3X1 매트릭스 선언
CMatrix Result;   // 빈 매트릭스 선언
A.Rand();         // A 매트릭스에 임의의 수로 채움
B.One();          // B 매트릭스에 1을 채움
Result=A.Invers()*B; // A의 역행렬과 B를 곱한
                     // 값을 결과매트릭스에 저장
```

위의 내용은 간단하게 예를 보여준 것이며 위의 예보다 더욱 복잡한 연산도 가능하다. 나머지 클래스는 다음 에서 자세히 다루도록 하겠다.

4. Execution and Control

아래의 그림은 최초 프로그램 실행시 제일 먼저 실행되는 다이얼로그 부분의 그림이다.



이 다이얼로그는 CControlDlg 라는 클래스가 담당을 하고 있으며 동역학 시뮬레이션의 핵심인 운동방정식과 프로그램의 제어 부분이 이 클래스 안에 들어있다. 여기서 나오는 결과를 가지고 COpenGLView Class가 화면에 모델의 움직임을 그리게 되는 것이다. CControlDlg 클래스 부분에서 실행되는 순서를 간단히 말로 설명하면,
Model Select 그룹에서 라디오 버튼을 클릭하게 되면 CVertex 클래스가 작동을 하여 라디오버튼이 눌려진 모델의 STL 파일에서 Vertex 정보를 추출해 오게 된다. 그뒤 사용자가 Start Time과

End Time을 알맞게 설정해 주고 Simulation 그룹의 Start 버튼을 누르게 되면 OnTimer() 함수가 호출이 되어 그 안에 시뮬레이션 코드를 실행하면서 동작하게 된다. 이때 프로그램의 모든 클래스가 서로 영향을 주면서 동시에 작동하게 된다. 시뮬레이션 도중에 Cur.Time 이라는 에디트 박스 에는 현재시간이 표시되면서 EndTime 까지 도달하게 되면 시뮬레이션이 멈추게 된다. 특히 위의 3가지 모델중에 Inverted Pendulum을 선택할 경우에는 오른쪽의 슬라이더 바를 이용하여 Pendulum을 제어 할수 있다.

5. Display

OpenGL을 이용하여 화면에 출력을 하는 부분의 코딩은 COpenGLView Class 가 담당한다고 설명 하였다. 초기 OpenGL 설정과 여러 마우스 이벤트에 따른 화면의 확대 및 축소 회전 이동 그리고 모델의 재질에 관련된 세팅은 공개된 MFC 템플릿을 인터넷에서 찾아서 만들었으며 나머지 부분은 직접 코딩하였다. 기본적으로 모델을 움직이게 할때에는 glMultMatrixf(*GLfloat); 함수를 사용하여 현재 모델의 상태에 트랜스포메이션 매트릭스를 계속 곱하여서 모델을 움직이도록 하였다. 이때 OpenGL의 트랜스포메이션 매트릭스는 열 매트릭스를 사용 하는 것이 아니라 행 매트릭스를 사용하여서 트랜스포메이션 매트릭스를 뒤에다 곱해 주어야 제대로 동작하게 된다. 크랭크와 같이 여러 관절이 있는 모델의 경우는 glPushMatrix(); glPopMatrix(); 사이에 여러번의 glPushMatrix(); glPopMatrix(); 를 집어넣어서 각 링크가 독립적으로 움직일 수 있도록 하였다.

6. Conclusion

결론으로는 이와 같이 OpenGL을 이용하여 프로그래밍을 하게 되면 비교적 쉽게 동역학 시스템을 시뮬레이션 해볼 수 있다. 또한 CMatrix 클래스와 CVertex 클래스를 이용하여 더욱 쉽고 빠르게 코딩 할수 있고 컴파일 시간도 단축 시킬수 있다.

7. References

[1] Catia로 배우는 CAD/CAM - 김재정

[2] Windows API 정복 - 가남사

[3] teach yourself C++ - 영진출판사

[4] Computer Aided Kinematics and Dynamics of Mechanical System Vol. 1 -Haug

[5] Introduction to Robotics - John J. Craig

[6] Wikipedia.org

*** 소 감**

학부 과정에서 배운 내용을 총정리 할 수 있어서 참 좋았습니다. 또 MFC 프로그래밍 실력이 이번 졸업 논문을 계기로 많이 향상된 것 같아서 굉장히 기분이 뿌듯합니다. 교수님의 많은 가르침 정말 감사드립니다.

http://www.filmlife.net/temp_june/Links/source.zip 에서 Source Code를 다운 받으실 수 있습니다.

*2018. 7. 9. 추가내용
filmlife.net 도메인 만료로 소스코드는 다음 주소로 이동
http://ikjunekim.net/index.files/thesis/code.zip