

오픈소스 물리엔진을 이용한 천장 크레인 시뮬레이터 개발*

옥수열[○], 김성길^{*}

동명대학교 게임공학과[○], 동명대학교 대학원 컴퓨터미디어공학^{*}
sooyol@tu.ac.kr, sisibo@nate.com

The Development of Overhead Crane Simulator Using Open Source Physics Engine

Soo-Yol Ok[○], Sung-Kil Kim^{*}

Dept. of Game Engineering, TongMyong University, Dept. of Computer Media Engineering, TongMyong University

요 약

최근 게임이나 시뮬레이터 개발에 있어서 물리 기반 시뮬레이션을 통해 사실성을 높인 고품질의 3D 콘텐츠를 개발하고자하는 사례가 많아지고 있다. 본 논문에서는 오픈소스 물리엔진인 ODE를 이용하여 중량물을 취급하는 천장크레인 시뮬레이터의 개발하는 방법을 제안하였다. 그리고 상용 천장크레인 시뮬레이터와의 물리적 특성 비교분석을 통해 오픈소스 물리엔진으로도 사실성 높은 훈련용 시뮬레이터 개발이 충분히 가능하다는 것을 확인하였다. 본 논문에서 제안한 천장크레인 시뮬레이터는 학교 등의 공공기관에서 교육용이나 훈련용으로 충분히 활용할 수 있을 것으로 예상된다.

ABSTRACT

Recently, increasing numbers of games and simulators are being implemented by employing the physically-based modeling techniques for better realism. In this paper, we propose the implementation techniques for overhead crane simulator based on ODE, the well-known open source dynamic engine. By comparing the dynamic behavior of the proposed system with a commercial engine based simulator, the physical plausibility and the effectiveness of the ODE based OHC simulator are verified. We expect the proposed the OHC simulator can be successfully utilized for virtual training in various educational institutes.

Keyword : Open Source, Overhead Crane Simulator, Physically Based Simulation, Open Dynamics Engine

접수일자 : 2009년 08월 13일

심사완료 : 2009년 09월 25일

* 이 연구는 지식경제부 및 정보통신산업진흥원의 대학IT연구센터 지원사업(NIPA-2009-C1090-0902-0004)의 연구결과로 수행되었음

1. 서 론

최근 컴퓨터 그래픽 기술의 발달과 함께 3D 게임 및 영화 등의 디지털 콘텐츠 시장에서 사실성과 현장감을 높이기 위한 다양한 형태의 기술개발이 이루어지고 있다. 그 중에서도 사실성 효과를 극대화하기 위해서 물리기반 시뮬레이션을 통해 표현하고자 하는 연구개발이 활발히 진행되고 있다 [1,2,3,4,5].

3D 콘텐츠 개발자들은 사실성이 높은 실감형 콘텐츠 제작을 위해 자체 물리엔진을 개발하여 적용하기 보다는 개발의 용이성 때문에 안정적인 상용 물리엔진을 사용하여 개발하는 경우가 많지만 중소 규모의 개발사나 학교와 같은 공공기관에서는 상용 엔진을 적용하기에는 비용적인 측면에서 어려운 형편이다.

이에 본 논문에서는 오픈소스 물리엔진인 Open Dynamics Engine(이하 ODE)[6]를 이용하여 하역 공정에서 중량물을 취급하기 위해 사용되는 천장 크레인(OverHead Crane, 이하 OHC) 시뮬레이터를 개발하는 방법을 제안한다.

또한 상용 OHC시뮬레이터와 제안한 시뮬레이터와의 물리적 특성 분석을 통해 오픈소스 물리엔진으로도 상용 물리엔진과 같은 사실성 높은 훈련용 시뮬레이터 개발의 가능성을 제시하고자 한다.

이를 통해 제안한 OHC시뮬레이터가 학교 등의 공공기관에서 교육용이나 훈련용으로 충분히 활용될 수 있을 것으로 예상된다.

본 논문의 구성은 다음과 같다. 먼저 2장에서 관련연구에 관해서 논하고, 3장에서는 제안 OHC시뮬레이터의 개발 방법에 대해서 설명한다. 4장에서는 제안한 OHC시뮬레이터의 실험결과를 설명하고, 마지막으로 결론에서는 본 연구의 타당성과 향후계획을 설명하고자 한다.

2. 오픈소스 물리엔진기반의 콘텐츠 개발

실감형 콘텐츠 제작에 있어서 물리엔진은 필요 불가결한 요소로 대두되고 있지만 상용엔진의 거의 대부분이 고가 제품으로 중소기업 및 학교 등과 같은 곳에서는 사용하기 힘들다. 또한 상용 엔진의 경우 사용자가 필요에 따라 새로운 기능을 추가하는 것이 어렵다.

[표 1] 오픈소스 공개용 물리 엔진의 예

엔진	라이선스	소스
ODE[6]	공개	O
Tokamak[7]	공개	O
Bullet[9]	공개	O
PAL[10]	공개	O
Dynamo[11]	공개	O
Physsim[12]	공개	O
OpenMASK[13]	공개	O

반면, [표 1]과 같은 오픈소스 물리엔진의 경우 무료로 엔진의 코드를 자유롭게 수정할 수 있어 오픈소스를 통한 다양한 시뮬레이터들의 개발이 이루어지고 있다[14,15]. 그 대표적인 오픈소스 물리엔진이 ODE이다. ODE는 안정적으로 강체의 동적 시뮬레이션이 가능하고, 또한 가상 운반기, 가상현실 환경의 객체, 가상 창조물 시뮬레이션 등에 유용하다[16].

[그림 1]은 ODE로 개발된 다양한 게임 및 시뮬레이터의 예를 나타내고 있다.



[그림 1] ODE로 개발한 게임 및 시뮬레이터

본 연구에서는 OHC시뮬레이터 개발을 위해 ODE의 주요 구성 요소 중 강체(Body)와 기하구조(Geometry) 그리고 조인트(Joint)를 이용하여 OHC시뮬레이터를 설계하였다. 강체에서는 질량 특성을 설정하고 기하구조에서는 객체의 형상 특징을 지정한다. 지정된 형상을 가지고 충돌처리 및 여러 물리적인 힘을 계산하며, 객체사이에 제한조건을 지정하기 위해서 조인트를 이용하여 움직임을 제어한다.

3. ODE를 활용한 천장 크레인 시뮬레이터 개발

본 연구에서 개발하고자 하는 천장 크레인은 항만이나 중공업 분야에서 하역공정에서 중량물을 취급하기 위해 많이 사용되고 있지만 실습을 통해 교육이나 연습할 수 있는 곳은 매우 드물다. 또한 실기 교육장을 만들기 위해서는 큰 규모의 장소뿐만 아니라 비용이 많이 드는 문제점이 있다.

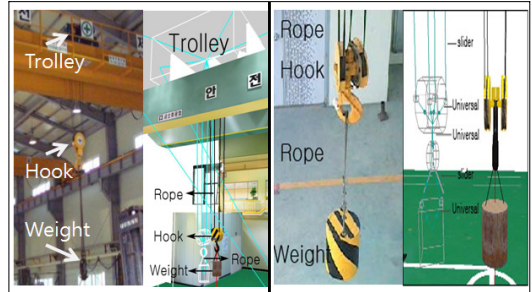
이에 본 논문에서는 [그림 2]와 같은 실기 운전 시험장의 천장 크레인을 모델[17]로 하여 교육 및 연습용으로 사용 가능한 사실성 높은 OHC시뮬레이터를 설계하고 시뮬레이션 하여 검증한다.



[그림 2] OHC 실기 운전 시험장의 모습

3.1 OHC시뮬레이터 구조 설계

3.1.1 크레인 구조 설계



[그림 3] 실기 OHC와 OHC시뮬레이터와의 구조비교

OHC시뮬레이터는 [그림 3]과 같이 맨 위에 트롤리(Trolley)가 놓여 있으며 트롤리와 후크(Hook)가 조인트로 연결되어 있다. 그리고 후크와 추(Weight)가 조인트로 연결되어 있다.

OHC시뮬레이터는 2개의 조이스틱을 양쪽으로 놓고 시뮬레이터를 조정한다. 왼쪽 조이스틱으로는 트롤리와 후크를 연결하는 조인트를 조절하여 추를 위 아래로 움직이는데 사용하고, 오른쪽 조이스틱으로는 트롤리를 전후좌우로 움직여 조인트로 연결된 후크와 추를 움직이는데 사용한다.

OHC시뮬레이터는 트롤리, 후크 사이드 앵커, 후크, 후크 하단 앵커, 추 사이드 앵커, 추로 구성되어 있으며 후크 사이드 앵커나 후크 하단 앵커, 추 사이드 앵커들은 객체들을 연결하기 위한 앵커들이다.

또한 트롤리와 후크 사이드 앵커는 슬라이더로 연결하여 상하로 움직이도록 하였고 후크 사이드 앵커와 후크를 고정하는 조인트는 유니버설 조인트로 정의하였다.

그리고 후크와 후크 하단 앵커를 유니버설 조인트로 연결하였고 후크 하단 앵커와 추 사이드 앵커를 슬라이더로 연결하였다. 마지막으로 추 사이드 앵커와 추를 유니버설 조인트로 연결하였다.

후크는 큰 실린더가 위에 있고 작은 실린더가 아래에 있으며 작은 실린더에 안테나처럼 박스 2

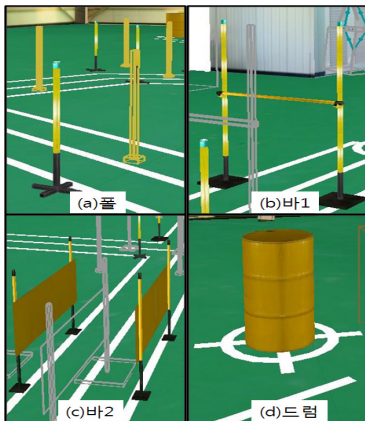
개가 나와 있다. 작은 실린더와 큰 실린더가 약간 떨어져 있지만 복합강체로 모델링하여 강체처럼 처리되기 때문에 아무런 문제가 없다.

3.1.2 장애물의 구조 설계

장애물 설계에 있어서도 실제 실기 시험에 사용되는 장애물을 모델로 하여 가상 장애물을 설계했다.

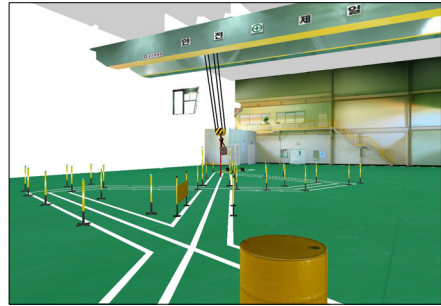
장애물의 종류로는 폴, 바1, 바2, 드럼으로 구성된다.

[그림 4]는 장애물들이 OHC시뮬레이터 상에서 3D 객체화된 형태를 나타내고 있다. [그림 4]에서 좌측 상단(a)은 폴이며 복합강체로 길다란 박스와 약간 납작한 박스로 설계했다. 우측 상단(b)은 바1의 모습이다. 바1도 복합강체이며 왼쪽, 오른쪽, 가운데의 박스는 길쭉하며 왼쪽 아래 오른쪽 아래 박스는 납작하게 구성되어 있다. 좌측 하단(c)은 바2이며 바1의 구조와 상당히 유사하지만 가운데 박스가 넓적하다. 드럼(d)은 복합강체가 아닌 그냥 박스로 이루어져있다. 후크와 추를 살펴보면 후크는 실린더 2개와 박스 2개로 구성되어 있으며 추는 박스 하나로 구성되어 있다.



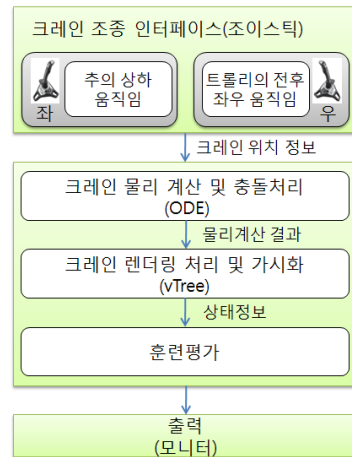
[그림 4] 장애물들의 구조

[그림 5]는 앞에서 설명한 내용을 바탕으로 하여 개발한 OHC시뮬레이터의 전체적인 시스템 이미지를 나타내고 있다.



[그림 5] 개발한 OHC시뮬레이터 이미지

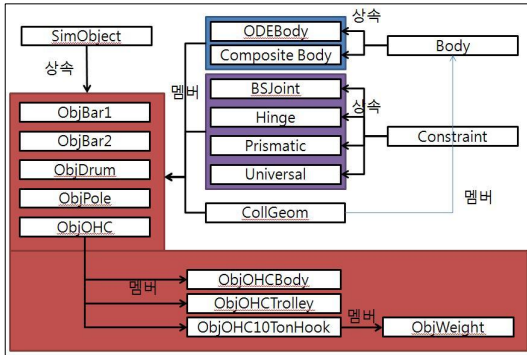
OHC시뮬레이터는 [그림 6]과 같이 데이터 처리 과정으로 이루어진다. OHC의 움직임이나 충돌 처리는 물리 엔진인 ODE에서 처리를 하며 계산된 결과는 렌더링 엔진인 VTree를 통해 화면에 출력된다. 그리고 장애물과의 충돌 횟수, 코스 라인을 벗어났는지를 검사하고 주행 시간들의 정보를 가지고 와서 훈련 평가를 하고 최종적으로 주행이 끝나거나, 충돌 시 체크되는 주요 실수(MajorError)가 5회가 넘어가거나 주행 시간이 7분 이상이 되면 훈련 결과에 대한 메시지를 출력한다.



[그림 6] OHC시뮬레이터 데이터 처리 과정

3.2 OHC시뮬레이터의 클래스 설계

OHC시뮬레이터는 [그림 7]과 같은 클래스의 구조로 설계하였다.



[그림 7] OHC시뮬레이터의 클래스 다이어그램

Body클래스는 ODE의 강체에 관련된 함수들로 구성되어 있다. 그 중 주요 함수들은 [표 2]와 같다.

AddForce함수는 강체에 힘을 가하는 함수이며 SetMass는 강체에 질량을 설정하는 함수이다. 단, 이 함수 안에서 바로 질량을 설정하지 않는다. 그 이유는 기하구조가 설정되지 않은 상태에서 강체에 질량이 들어가면 ODE에서 에러를 발생하기 때문이다.

MakeKinematic은 강체를 키네마틱 객체로 만들어서 물리적으로 동작하지 않게 만드는 함수이며 MakeDynamic은 강체를 물리적으로 동작하게 하는 함수이다.

CompositeBody클래스는 복합강체 객체를 만들기 위한 클래스이다. 객체가 복합강체일 경우 새로운 객체를 추가로 넣는 AddCollGeom함수 부분을 제외하고는 Body클래스와 같다.

AddCollGeom함수는 ODE의 기하구조 정보를 기존의 기하구조에 새로운 기하구조를 추가하는 역할을 한다.

[표 2] Body클래스의 주요 함수

Body	
함수명	파라미터
AddForce	float fx, fy, fz
SetMass	double mass
MakeKinematic	
MakeDynamic	
CompositeBody	
함수명	파라미터
AddCollGeom	CCollGeom* Geom
ODEBody	
함수명	파라미터
SetCollGeom	SetCollGeom* Geom

CollGeometry 클래스는 객체의 형태를 표현하기 위한 클래스로 Body클래스와 연결하여 객체간 충돌처리를 할 수 있다. CollGeometry클래스는 [표 3]과 같은 주요 함수들로 구성 되어 있다.

[표 3] CollGeometry클래스의 주요 함수

CollGeom 클래스	
함수명	파라미터
CreateBox	float dx, dy, dz
CreateSphere	float radius
CreateCylinder	float radius, height
CreateTriangle Mesh	int VerticeCount, TriangleCount VECTOR3f *Vertice, VECTOR3f *TriangleIndex

CreateTriangleMesh는 3D의 메쉬 파일을 읽어와 기하구조를 생성한 다음 질량을 설정한다.

Constraint클래스는 조인트를 연결하는 클래스로 주로 어떤 객체와 어떤 객체를 무슨 조인트로 연결한 것인지 앵커의 위치를 어디로 할 것인지 설정하는 클래스이다. Constraint클래스는 [표 4]와 같은 주요 함수들로 구성되어 있다.

[표 4] Constraint클래스 관련 주요 함수

Constraint 클래스	
함수명	파라미터
SetBodies	CBody * FirstBody, CBody *SecondBody
GetBodies	CBody * FirstBody, CBody *SecondBody
SetAxis	float Axis[3]
SetAxes	float Axis1[3], float Axis2[3]
GetAxis	float Axis[3]
GetAxes	float Axis1[3], float Axis2[3]
SetUpperLimit	float Limit
SetLowerLimit	float Limit

SetAxis는 조인트에 축 설정을 의미하는데 힌지(Hinge)처럼 회전축을 y축을 주면 y축이외의 축은 움직이지 않는다. 슬라이더(Slider) 축은 경우에는 축이 y인 경우 위, 아래로만 움직인다. 그리고 볼-소켓 조인트처럼 축이 없는 경우의 유니버설처럼 축이 2개인 조인트로 나누어서 함수를 설계하였다.

Constraint클래스에서 조인트와 관련된 함수들은 파생클래스로 하여 관련 클래스들을 설계하였다.

OHC시뮬레이터에는 [표 5]과 같이 여러 객체들을 실기 OHC구조에 맞게 객체를 구성했다.

[표 5] 시뮬레이터에 사용된 객체

환경	객체	구조특성	구성
크레인	Trolley	Body	Box 1개
	Weight	Body	Box 1개
	Hook	Composite Body	Box 2개 Cylinder 2개
장애물	Pole	Composite Body	Box 2개
	Bar1	Composite Body	Box 5개
	Bar2	Composite Body	Box 5개
	Drum	Body	Box 1개

3.3. 시뮬레이터의 환경 설정

3.3.1 질량 설정

ODE는 기하구조가 없는 상태에서 질량을 설정하면 문제가 생긴다. 그 이유는 ODE에서 질량을 설정할 때 기하구조의 중심에 질량을 설정하기 때문이다. 따라서 강체만 생성해서 질량을 설정되는 것이 아니라 바디에 지오메트리를 연결시킨 뒤에 질량을 설정할 필요가 있다.

복합강체의 경우는 기하구조를 한 강체에 여러 개를 중첩시켜서 만드는 형태로 기존의 기하구조에 새로운 기하구조를 중첩할 때마다 질량의 위치를 변경된 기하구조의 중심으로 이동시켜줘야 한다. 질량의 위치가 잘못 이동되어 무게 중심이 아래에 있을 경우 객체가 넘어졌다가 다시 일어서는 오펜이(tumbler) 현상이 일어날 수도 있다.

3.3.2 충돌 처리

OHC시뮬레이터는 크레인과 장애물 사이에서의 충돌처리를 통해 훈련의 성공여부를 판단한다. 충돌처리를 할 때 충돌 가능성이 있는 2개의 기하구조 객체를 조인트로 연결해서 물리적인 계산으로 충돌처리를 한다.

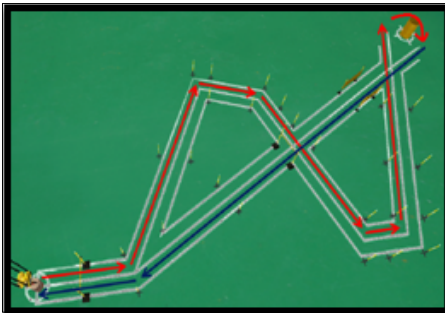
이때 공간 내에서 잠재적으로 충돌할 수 있는 기하 구조 쌍을 결정하고 각 후보 쌍에 대해 함수를 호출하는데, 가끔씩 충돌이 일어나지 않았는데도 불구하고 충돌처리를 하는 경우가 발생한다. 이런 문제를 방지하기 위해 충돌처리 함수를 호출할 때 먼저 충돌지점이 몇 개인지 검사한 후 충돌지점이 1개 이상인 경우 충돌처리를 하도록 하였다.

또한 충돌처리 고속화를 위해 크레인과 장애물 간만을 충돌 처리하도록 하였다. 즉 충돌 시 충돌한 객체들의 이름을 가져와서 한쪽 이름이 후크나 추이고 다른 한쪽이 폴, 바1, 바2인 경우 충돌 사운드를 재생하고 MajorError를 증가 시키며 이 MajorError가 5이상 넘어가면 훈련 실패 메시지를 화면에 출력한다.

3.3.3 OHC시뮬레이터 훈련 코스

[그림 8]은 OHC시뮬레이터에서 크레인 운전 훈련을 코스로서 실제 환경과 동일하게 설계했다.

주행시간 7분 이내로 돌아야 하며 빨간색 선을 따라서 간 다음 드럼 위에 추를 올려놓고 다시 파란색을 따라서 원점으로 돌아오면 된다. 원점으로 돌아오면 결과창과 함께 시뮬레이터는 종료가 된다.



[그림 8] OHC시뮬레이터 훈련 코스 환경

[그림 9]는 사용자가 훈련 코스를 제대로 돌아서 원점으로 돌아왔을 시 화면에 훈련결과를 알려주는 화면이다. 훈련 결과는 성공인지 실패인지 출력하고 주행시간과 충돌 시 발생하는 MajorError와 코스 라인을 벗어날 때 일어나는 MinorError 그리고 훈련명이 나타난다.



[그림 9] 훈련 결과 창

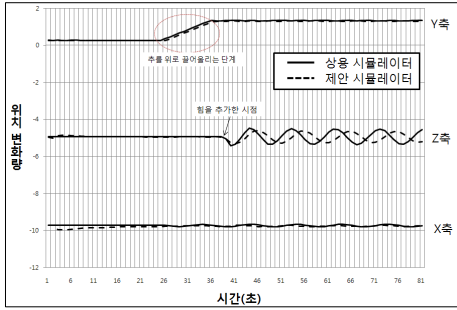
4. 실험 결과 및 분석

본 논문에서는 개발한 OHC시뮬레이터를 [그림 10]과 같이 운전 실험을 통하여 성능 검증을 하였다. 본 실험에서는 실기 OHC와의 비교 분석에 있어서 실기 OHC의 물리적 특성 측정에 어려움이 있어, TSB사에서 상용 물리 엔진인 Vortex엔진 [19]으로 개발하여 현장테스트를 걸쳐 교육 및 훈련용으로 사용하고 있는 상용 시뮬레이터와 비교 실험을 하였다.



[그림 10] 개발한 OHC시뮬레이터의 운전 실험 모습

성능 비교 방법은 후크에 동일한 힘을 주어 흔들림의 물리적인 특성이 어느 정도 유사한지를 비교해보았다[20,21,22]. 힘은 후크의 Z축 방향으로 500만큼 주어 시계추와 같이 흔들리게 하여 그 변화량을 비교해보았다. [그림 11]은 후크의 흔들림에 따라 추의 X, Y, Z축에 대한 변화량을 0.5초 단위로 나타낸 그래프이다. 여기서 X는 좌우, Y는 높이, Z는 전후의 움직임을 나타낸다. 추의 초기 시작위치는 기준 좌표계상에서 각각 X는 -9.7, Y는 0.2, Z는 -4.9로 설정하여 실험하였다.

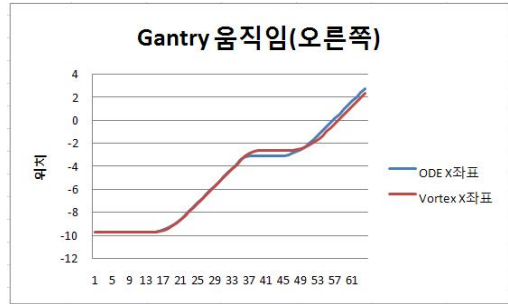


[그림 11] 추 변위의 변화량 비교 그래프

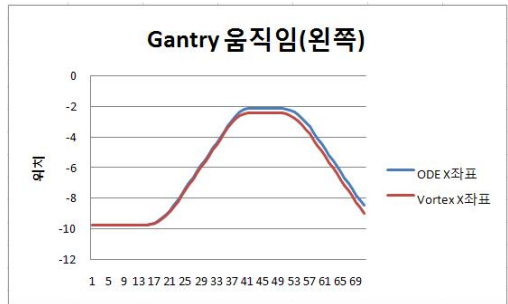
[그림 11]에서 나타난 것과 같이 본 연구에서 개발한 OHC시뮬레이터가 TBS사의 상용시뮬레이터와 비교해서 큰 차이가 없다는 것을 할 수 있다. 전후 흔들림의 변화량을 나타내는 Z축의 그래프 에서 상용 시뮬레이터에 비해 개발 시뮬레이터가 물리적인 흔들림에 있어서 약간의 차이가 발생함을 알 수 있다. 이는 상용 시뮬레이터가 제안한 시뮬레이터보다 조금 빨리 진동을 하는 차이만 있을 뿐 주거나 진폭의 크기는 거의 비슷하다는 것을 알 수 있다.

[그림 12,13]은 조이스틱으로 겐트리(Gantry)를 좌우로 조종했을 때 겐트리의 움직임 변화를 나타낸 것이다. [그림 12]는 겐트리를 중앙에 위치시켜 놓고 조이스틱으로 오른쪽으로 속도를 높였을 때 겐트리의 위치 변화를 나타내고 있다.

또한 [그림 13]은 왼쪽으로 속도를 높였을 때 겐트리의 위치 변화를 나타내고 있다. 본 논문에서 제안한 시뮬레이터가 [그림 12,13]에서 알 수 있듯이 상용물리엔진과 미소한 차이는 있지만 거의 동일하다는 걸 확인 할 수 있다.

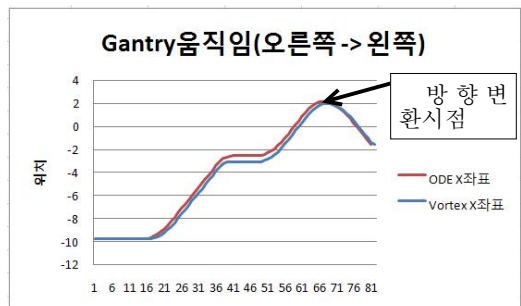


[그림 12] 오른쪽 방향으로 속도 증가시 겐트리의 움직임 변화그래프



[그림 13] 왼쪽 방향으로 속도 증가시 겐트리의 움직임 변화그래프

또한 [그림 14]는 조이스틱으로 겐트리를 오른쪽으로 이동하다가 왼쪽으로 방향전환을 했을 때 겐트리의 움직임의 변화를 나타낸 것이다.



[그림 14] 좌우 방향전환시의 겐트리의 움직임 변화그래프

겐트리의 방향전환 실험에서도 제안 시뮬레이터

가 상용물리엔진을 이용한 시뮬레이터와 비교해서 거의 유사한 움직임을 나타내고 있는 것을 확인할 수 있다. 다만, 미소한 차이는 매개변수조절로 충분히 해결될 것으로 예상된다.

본 논문에서 제안한 시뮬레이터가 상용물리엔진 시뮬레이터와 비교해 거의 동일하다는 것을 주요 특성 실험을 통해 확인할 수 있었다. 따라서 제안한 OHC시뮬레이터가 학교 등의 공공기관에서 교육용이나 훈련용으로 충분히 활용될 수 있을 것이다.

5. 결 론

본 논문에서는 오픈소스 물리엔진인 ODE를 이용하여 항만이나 중공업 분야의 생산 및 하역공정에서 중량물을 취급하기 위해 사용되는 OHC시뮬레이터의 개발방법을 제시하였다. 특히, 제안한 시뮬레이터와 상용 OHC시뮬레이터와의 물리적 특성 비교 실험을 통해 오픈소스 물리엔진으로도 상용 물리엔진과 같은 사실성 높은 훈련용 시뮬레이터 개발이 가능하다는 것을 확인하였다.

제안한 시뮬레이터가 상용 시뮬레이터에 비해 조금 빨리 진동을 하는 차이만 있을 뿐 물리적인 특성 차이가 거의 없을 정도의 고품질의 사실적인 시뮬레이터를 개발할 수 있다는 것을 확인하였다.

향후 본 연구에서는 개발한 OHC시뮬레이터를 실기 OHC의 물리적인 특성 데이터와 비교 검증을 통해 OHC시뮬레이터의 사실적인 정밀도를 높이고자 한다. 이를 통해 개선된 OHC시뮬레이터는 학교 등의 공공기관에서 교육용이나 훈련용으로 충분히 활용할 수 있을 것으로 예상된다.

참고문헌

- [1] Eran Guendelman, Andrew Selle, Frank Losasso, Ronald Fedkiw, "Coupling water and smoke to thin deformable and rigid shells", ACM Transactions on Graphics, 24(3), pp. 973-981, 2005.
- [2] Danny M. Kaufman, Shinjiro Sueda, Doug L. James, Dinesh K. Pai, "Staggered Projections for Frictional Contact in Multibody Systems", ACM Transactions on Graphics, 27(5), pp. 164:1-164:11, 2008.
- [3] Christopher D. Twigg, Doug L. James, "Many-Worlds Browsing for Control of Multibody Dynamics", ACM Transactions on Graphics, 26(3), pp. 14:1-14:8, 2007.
- [4] Mark Carlson, Peter J. Mucha, Greg Turk, "Rigid Fluid: Animating the Interplay Between Rigid Bodies and Fluid", ACM Transactions on Graphics, 23(3), pp. 377-384, 2004.
- [5] J. Georgii and R. Westermann, "Corotated Finite Elements Made Fast and Stable", VRIPHYS Workshop in Virtual Reality Interactions and Physical Simulations, 2008.
- [6] ODE website, <http://ode.org/>
- [7] Tokamak website, <http://www.tokamakphys.ics.com>
- [8] PhysX website, http://www.nvidia.com/object/nvidia_physx.html
- [9] Bullet website, <http://www.bulletphysics.com/Bullet/wordpress/>
- [10] PAL website, <http://www.adrianboeing.com/pal/index.html>
- [11] Dynamo website, <http://home.claranet.nl/users/starcat/dynamo>
- [12] Physsim website, <http://physsim.sourceforge.net/>
- [13] OpenMASK website, <http://www.irisa.fr/bunraku/OpenMASK/index.htm>
- [14] Delta3D website, <http://www.delta3d.org/>
- [15] Perry McDowell, Rudolph Darken, Joe Sullivan, Erik Johnson, Delta3D: A Complete Open Source Game and Simulation Engine for Building Military Training Systems, JDMS 3(3). pp. 143-153, 2005.
- [16] Bing Tang, Zhigeng Pan, ZuoYan Lin, and Le Zheng, "PHI: Physics Application

- Programming Interface”, ICEC 2006, LNCS 4161, pp 390-393, 2006.
- [17] 천장크레인 실기 훈련 website, <http://www.helheavy.com/07cranel/crane01.htm>
- [18] Vtree website, <http://www.vtreeinc.com/>
- [19] Vortex Physics website, <http://www.vxsim.com/>
- [20] 이호훈, 조성근, “천정크레인 부하의 위치 및 흔들림 제어”, 대한기계학회논문집 A권 제21권 제2호, pp. 297-304, 1997.
- [21] 박병석, 김준홍, 윤지섭, 강이석, “산업용 천정 크레인의 이송위치 및 이송물 흔들림 제어에 관한 연구”, 한국정밀공학회 추계학술대회 논문집(I), pp.15-18, 1999.
- [22] 배상욱, 노철균, 배영호, 이득기, “천정크레인 설비의 자동화를 위한 반진동 제어 알고리즘”, 조명·전기설비학회논문지 제17권 제2호, pp.49-57, 2003.
- [23] 옥수열, 김성길, “게임 물리엔진을 이용한 천장 크레인 시뮬레이터 개발”, 한국자동제어학회회의(KACC2009), pp. 378-383, 2009



옥 수 열(Soo Yol, OK)

1998년: 쑤쿠바대학 이공학연구과 공학석사
2001년: 쑤쿠바대학 공학연구과 공학박사
2001년~2004년: 일본 통신종합연구소 연구원
2004년~현재: 동명대학교 정보통신대학 미디어공학부 게임공학과

관심분야 : 컴퓨터 시뮬레이션, 게임 인공지능, 컴퓨터 그래픽스, 가상현실



김 성 길(Sung Kil, Kim)

2009년: 동명대학교 게임공학과 학사
현재 동명대학교 대학원 컴퓨터미디어공학과 석사과정재학

관심분야 : 컴퓨터 시뮬레이션, 게임 인공지능, 게임 알고리즘.