

---

# 물리 기반 캐릭터 애니메이션을 위한 반응 모션 생성 기법

## Animating Reactive Motions for Physics-Based Character Animation

지현호, HyunHo Jee\*, 한정현, JungHyun Han\*\*

---

**요약** 컴퓨터 게임 및 가상현실에서 가상 캐릭터와의 상호작용은 매우 중요한 문제이다. 본 연구는 외력이 인간형 캐릭터에 적용될 때 반응 모션을 생성하는 기법을 제안한다. 이 기법에서는 외력과 인간형 캐릭터의 상태를 고려하여 이동할 다리를 선정하고, 두 다리가 교차하지 않는 범위 내에서 몸의 균형을 유지시켜줄 발의 목적 위치를 결정한다. 다음으로 다리 관절 제어를 통해 발을 들고 제어 메커니즘을 통해 반응 모션을 생성한다. 본 연구는 기존의 방식과는 다르게 미리 캡처된 반응 모션 데이터를 사용하지 않으면서도 캐릭터 상황에 맞는 자연스러운 반응 모션의 생성을 보여준다.

**Abstract** The technique for synthesizing reactive motion in real-time is important in many applications such as computer games and virtual reality. This paper presents a dynamic motion control technique for creating reactive motions in a physically based character animation system. The leg to move in the next step is chosen using the direction of external disturbance forces and states of human figures and then is lifted though joint PD control. We decide the target position of the foot to balance the body without leg cross. Finally, control mechanism is used to generate reactive motion. The advantage of our method is that it is possible to generate reactive animations without example motions.

**핵심어:** *Character Animation, Physics-Based Animation, Ragdoll, Motion Control*

---

이 논문은 2007년도 정부재원(문화관광부)으로 한국게임산업진흥원의 지원을 받아 연구되었음.

\*주저자 : 고려대학교 컴퓨터학과 석사과정

\*\*교신저자 : 고려대학교 컴퓨터학과 교수; e-mail: jhan@korea.ac.kr

## 1. 서론

많은 연구자들은 가상 인체 모델의 자연스러운 모션 생성을 연구해왔다. 이러한 모션을 만들어내는 것은 영화, 가상 환경, 게임 분야와 같은 다양한 어플리케이션에서 중요한 연구 분야이다. 최근 연구자들은 인터랙티브한 모션을 요구하는 많은 응용 분야에 적합한 알고리즘 개발에 큰 관심을 가지고 있다. 사람의 모션을 합성하는 방법은 크게 두 가지 연구 방향이 있다. 첫 번째는 필요한 행동이 담겨있는 방대한 양의 모션 데이터를 수집하고 이를 바탕으로 재구성하는 방법이고, 두 번째는 역운동학(inverse kinematics)과 물리적 시뮬레이션(physical simulation)을 기반으로 모터 제어 기술을 이용해 모션을 생성하는 방법이다. 전자는 다양한 환경 변화에 의한 반응 모션을 생성하기 위해서는 엄청난 양의 반응 모션 데이터를 필요로 하고, 상황에 맞는 모션을 찾는 데 많은 시간이 소요되므로 실시간 프로그램에는 적합하지 않다는 단점이 있다. 반면 후자의 방법은 물리 기반 시뮬레이션을 통해 실제 사람의 움직임 데이터를 녹화하기 어려운 위험한 스텐트 동작 즉, 떨어지거나 폭발에 날아가는 역동적인 동작을 효율적으로 제작할 수 있는 장점이 있다.

최근 컴퓨터 게임에서는 캐릭터와 환경과의 상호작용을 하는 랙돌(ragdoll) 시뮬레이션을 많이 이용하지만 살아있는 사람의 자연스러운 행동을 제어하는 시스템이 없기 때문에 시체와 같은 움직임만을 보인다. 이를 극복하기 위해 제한된 몇 개의 연구 방법은 외력의 작용에 대한 자연스러운 반응을 고려하고 있다.

본 연구에서는 때리고 밀고 당기는 것과 같은 외부의 작용에 대해 반응하는 모션을 생성하는 새로운 방법을 제안한다. 캐릭터는 이러한 다양한 조건의 외부 작용에 의해 균형을 잃게 되는데, 제어 메카니즘을 사용하여 두 다리가 서로 충돌하지 않는 범위 내에서 발의 내디딤을 통해 균형을 회복한다. 캡처된 반응 모션 데이터의 활용 없이도 자연스러운 반응 모션을 생성한다. 또한 캐릭터가 서있거나 뛰거나 달리는 등, 두 발의 지면과의 접촉 유무가 다른 모션에 적용이 가능하다. 본 연구는 실제 사람에게 외력을 수차례 주어 그에 대한 반응 행동을 관찰하고 이를 분석한 결과를 바탕으로 한다. 입력 모션으로는 모션 캡처 데이터와 같은 운동학 제어 기술(kinematic control techniques)을 사용하는데, 이미 실제적이고 물리적으로 안정적이라 가정한다. 외력이 가해졌을 때에 제어를 통해 반응 모션을 생성하고 다시 운동학 제어 기술로 전이한다. 본 연구를 통해 생성된 반응 모션은 물리적으로 정확하기 보다는 적은 연산으로도 사람 모션에 유사하도록 시뮬레이션하는 것에 중점을 둔다.

본 논문은 다음과 같이 구성된다. 먼저, 2장에서는 관련 연구를 소개하고, 3장에서는 물리 시뮬레이션을 위한 인체의 구성을 설명하며, 4장에서는 반응 모션을 생성하는 단계를

제시한다. 그리고 5장에서는 구현 및 결과를, 그리고 6장에서는 결론을 서술한다.

## 2. 관련 연구

학술적인 측면에서 볼 때, 이족보행 인체 모형의 물리 기반 시뮬레이션 및 제어는 대표적인 난제 중 하나이다. 물리 기반으로 모션을 생성하고 수정하는 방법은 크게 공간시간 제약조건(spacetime constraints)과 동적 시뮬레이션(dynamic simulation) 방법이 있다. 공간시간 제약조건은 지정된 제약 조건을 만족시키면서 최적화된 캐릭터의 동작을 자동으로 생성하기 위한 방법이다. [1]에서는 공간시간 제약조건을 이용한 변형 기법을 제안하였다. 비록 공간시간 제약조건 방법이 물리적으로 사실적이고 최적화된 동작을 자동으로 생성해줄지라도 최적의 해를 구하는데 필요한 연산량이 매우 많고, 환경과 상호작용하는 모션을 생성할 수 없다는 단점이 있다.

동적 시뮬레이션 방법은 관절 토크(torque)를 계산하기 위해 캐릭터의 현재 상태와 목적 모션을 기반으로한 동적 제어기(dynamic controller)를 사용한다. 이 토크를 기반으로 전방 동역학 시뮬레이션(forward dynamics simulation)은 결과 모션을 생성한다. 많은 연구자들은 걷거나 운동 경기에서 사용되는 움직임과 같은 캐릭터의 특정 행동을 제어기를 사용해서 개발했다[2, 3]. 하지만 캐릭터가 외력을 받았을 때 효과적으로 대처하지 못하는 단점이 있다.

관련 연구로 [4]는 여러 제어기들을 구성하고 사용하는 프레임워크를 제안하였고, [5]는 동적 환경에서 캐릭터가 편치에 맞고 부드럽게 원상 복귀하는 효과를 PD-controller를 사용하여 구현하였다. 또한 [6]은 역운동학을 이용하여 짐이 캐릭터의 등에 갑자기 붙는 효과와 같은 반응 모션의 생성을 보였다. 비록 이러한 방법들은 환경과의 상호작용을 고려하고 있지만 걷거나 뛰는 것과 같이 주기적으로 발의 접촉 패턴이 변하는 모션에 대해서는 고려하고 있지 않다.

캐릭터에 때리고 밀고 당기는 것과 같은 외력을 가했을 때, 발의 내딛는 동작을 통하여 균형을 유지하는 반응 모션의 생성을 시도한 연구도 있다. [7]은 운동량기반의 역운동학과 데이터베이스에 저장된 반응 모션의 검색을 통해 문제를 해결하고자 하였고, [8]은 유한상태기계(finite state machine)와 관절 제어를 통해 반응 모션을 생성하였다. 전자의 경우 검색에 추가적인 계산을 요구되고, 후자의 경우 다리 충돌에 대한 고려가 없다는 단점이 있다.

운동학 애니메이션과 물리 시뮬레이션의 합성을 통해 이들의 장점을 모두 살리는 연구도 활발히 진행되고 있다. [6]에서는 캐릭터가 입력 모션을 따라 움직이다가 외력이 가해졌을 때 균형 유지를 위한 제어기를 사용하였고, [9]에서는

물리적으로 생성된 모션과 데이터베이스에 있는 반응 모션의 합성을 통하여 새로운 모션을 생성했다. 본 연구는 [10]에서 사용한 결합 기술을 사용한다.

### 3. 물리 시뮬레이션 인체 모델

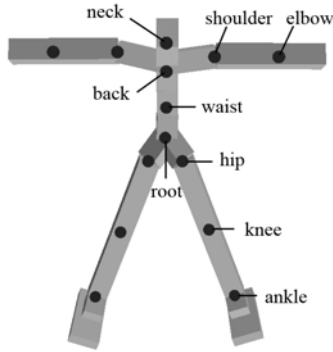


그림 1. 시뮬레이션에서의 인체 모델 구성

본 연구의 가상 인체 모델은 관절로 연결된 강체(rigid body)들의 집합으로서 관절체(articulated body)로 모델링된다. 시뮬레이션 모델은 14개의 관절로 구성되어 있고 각 관절은 1 또는 2의 자유도(degrees of freedom)를 가진다. 그림 1은 시뮬레이션 모델을 나타낸다.

표 1. 관절의 자유도와 제약사항 목록

Joint	DOF	Low <sub>x</sub>	High <sub>x</sub>	Low <sub>y</sub>	High <sub>y</sub>
Neck	2	-20.0	20.0	-60.0	60.0
Shoulder	2	-100.0	30.0	-120.0	100.0
Elbow	1	0.0	120.0	-	-
Back	1	0.0	0.0	-	-
Waist	1	-2.5	5.0	-	-
Hip	2	-90.0	30.0	-80.0	20.0
Knee	1	0.0	160.0	-	-
Ankle	2	-60.0	60.0	-40.0	40.0

관절의 구속은 물리적으로 인체 모델 각 관절의 움직일 수 있는 범위를 제약한다. 각 관절의 자유도에 상한과 하한 값을 설정한다. 관절의 범위는 생물역학적인 자료[11]를 기반으로 정하였다. 표 1은 각 관절의 자유도와 한계 값을 나타낸다. 또한 강체로 이루어진 인체의 각 부위의 질량은 생물역학 데이터[3]에 근거하여 정하였다. 하지만 신체 구조와 무게, 관성(inertia) 등 물리적 성질은 정확하게 측정하기 대단히 어렵다. 따라서 본 연구에서는 정확한 인체의 물리 시뮬레이션 보다는 보기에 자연스러운 모션 생성에 중점을 둔다.

#### 3.1 인체 표현

입력 모션은 모션 캡처 데이터나 키프레임 애니메이션을 사용한다. 이러한 운동학 모델(kinematic model)은 중심 링크(link)의 위치와 부모 링크 좌표계의 상대적인 방위로 다음과 같이 나타낸다.

$$M(t) = \{p(t), q_1(t), \dots, q_k(t)\} \quad (1)$$

$M(t)$ 는  $t$ 에서의 모션을 나타내고,  $p$ 는 중심의 위치,  $q_j$ 는  $j$ 번째 관절에서 부모 링크로부터의 상대 방위를 나타낸다. 계층 구조는 관절 제어기에 원하는 각을 입력하기 좋고, 전역 절대 위치 정보는 외력에 대한 처리를 하는 것이 좋으므로 상호 변환이 가능하게 한다.

#### 3.2 제어 시뮬레이션

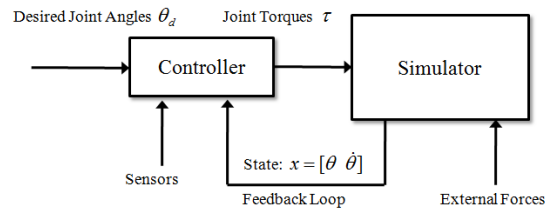


그림 2. PD-controller를 사용하는 기본 피드백 시스템 구성도

제어기는 가상 인체의 관절을 움직이기 위한 간단한 메카니즘(mechanism)을 제공한다. 그것은 관절 내부의 토크를 계산하고 적용하여 실제 사람의 근육과 같은 역할을 해준다. 제어 시스템의 기본은 관절의 목적 위치를 설정하고 관절 토크를 사용하여 현재의 관절이 목적 위치로 이동하도록 계산하는 것이다. 제어기의 토크 계산은 proportional-derivative controller(PD-controller)를 사용하여 구현한다. PD-controller는 식 (2)와 같다.

$$\tau = k_p(\theta_d - \theta) - k_v\dot{\theta} \quad (2)$$

$\theta$ 와  $\theta_d$ 는 각각 관절의 현재 각도와 목적 각도를 나타낸다.  $\dot{\theta}$ 는 관절의 현재 각속력,  $k_p$ 와  $k_v$ 는 각각 위치와 속력의 조절 상수 값을 나타낸다. 출력되는  $\tau$ 는 관절 내부의 토크를 나타낸다. 안정적이고 자연스러운 모션 생성을 위해서는 상수 값의 선정이 중요하데, 일반적으로 수식을 통하기 보다는 경험적으로 얻어진 적당한 값을 적용했다. 하지만 [2]에서는 이러한 파라미터들의 값을 최적화하는 기술을 개발했다. 그림 2는 PD-controller를 사용하는 기본적인 피드백 제어 시스템을 나타낸다. 시스템은 제어기와 시뮬레이터의 두 개의 모듈로 구성된다. 각 시뮬레이션 단계마다 제어

기는 캐릭터의 현재 상태와 사용자의 입력을 바탕으로 토크를 계산한다. 시뮬레이터에서는 제어를 통해 받은 토크와 외력을 기반으로 캐릭터의 상태를 갱신한다.

#### 4. 반응 모션 생성

몸의 균형을 유지하고 걷고 달리는 등의 실제 사람의 움직임은 복잡한 근육의 제어를 통하여 이루어진다. 이러한 행동 메커니즘을 분석하고 시뮬레이션하기란 상당히 어렵다. 따라서 본 논문에서는 단순한 몇 가지 효율적인 제어 방식을 채택하여 실제 사람의 행동에 가까운 반응 모션을 생성한다. 반응 모션은 그림 3의 과정을 거쳐 생성한다.

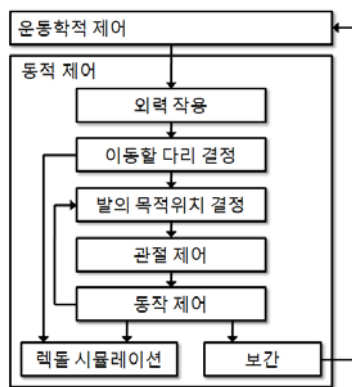


그림 3. 반응모션 생성 흐름도

일단 외력이 작용하기 전의 입력 모션은 걷고 뛰고 서있는 등의 두 발의 지면과의 접촉 여부가 다른 운동학적 모델 데이터를 사용한다. 발의 지면과의 접촉 여부는 전처리기에서 처리된 정보를 사용한다. 사용자는 외력의 방향과 크기를 지정하여 캐릭터의 특정 부위에 가한다. 외력은 수평 방향으로만 작용한다고 가정한다. 반응 모션이 생성될 동안의 상체는 외력이 있기 전의 자세를 유지한다. 이를 위해 외력을 가하기 전의 상체 정보를 저장하고, 시뮬레이션 하는 동안 PD-controller를 사용하여 상체들의 관절각을 유지한다. 이 절을 통해 외력이 작용한 후의 반응 모션 생성 단계를 보인 다.

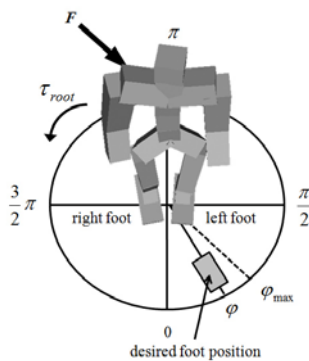


그림 4. 발의 목적위치 결정

#### 4.1 이동할 다리 결정

이동할 다리는 외력의 방향과 두 발의 지면과의 접촉 여부에 따라 결정한다. 두 발이 모두 지면과 접촉해 있을 때 외력을 받으면 캐릭터는 그 진행방향으로 기울어지게 된다. 기울어진 방향의 안쪽 다리는 몸을 지탱하게 되고 외력을 작용한 방향의 다리는 자연스럽게 들리기 때문에 바깥쪽 다리를 다음에 이동할 다리로 선정한다. 예를 들어 그림 4의 캐릭터는 두 발이 모두 지면에 닿아있고 외력은 오른쪽 방향 어깨의 뒤쪽에서 작용되었기 때문에 이동할 다리로 오른쪽을 선정한다. 반면 걷기와 달리기와 같은 동작은 한쪽 다리만 지면과 접촉해 있을 경우가 있다. 이때는 이미 들린 다리를 다음에 이동할 다리로 선정한다. 두 발이 모두 지면에 닿아 있지 않은 경우는 렉들 시뮬레이션을 적용하여 캐릭터를 자연스럽게 넘어지도록 한다.

#### 4.2 발의 목적위치 결정

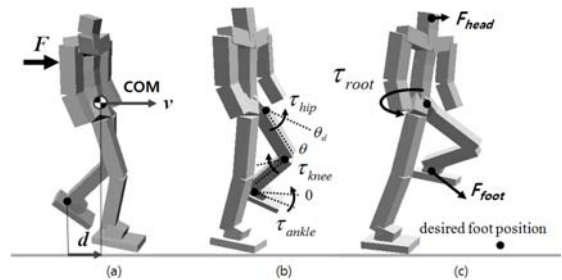


그림 5. 반응 모션 생성 전략: (a) 무게 중심의 위치와 속도 그리고 외력을 이용한 발의 목적 위치 결정, (b) 다리 관절들의 현재 각과 목적 각을 통한 다리 제어, (c) 추가적인 외력에 의한 캐릭터의 이동 제어

이동할 다리가 결정된 후에는 인체가 균형을 잡기 위하여 발이 놓여야할 목적위치를 결정한다. 발의 목적위치는 처음 외력을 받았을 때는 식 (3)를 이용하여 구하고 다음 반복부터는 식 (4)를 이용한다. 동작 제어 단계에서는 중심의 속도가 일정 값 보다 작아지면 캐릭터가 균형을 모두 회복한 것으로 판단하고 발의 목적 위치를 운동학적 모델의 발의 위치로 정한다. 즉, 운동학적 제어 단계로의 자연스러운 전이를 위해 지면에 닿아 있는 발을 중심으로 지정된 운동학적 모델을 정렬하고, 반대편 발의 위치를 목적위치로 정한다.

$$D_0 = \lambda (COM + c_v v + c_f F) \quad (3)$$

$$D_{k-1} = \lambda (COM + c_v v + c_d d) \quad (4)$$

$D$ 는 이동하는 발의 목적위치를 나타내고,  $\lambda$ 는 벡터 (vector)의  $y$ 축 성분을 지면으로 투영시켜주는 함수이다.  $COM$ 는 인체의 무게 중심(center of mass)의 위치를,  $v$ 는

무게 중심의 속도를 나타낸다.  $F$ 는 외력을,  $d$ 는 무게 중심에서 멀리 떨어진 발에서부터 무게 중심까지의 방향을 나타낸다.  $c_f, c_d, c_v$ 는 각각  $F$ 와  $d$  그리고  $v$ 의 계수를 나타낸다. 그림 5의 (a)는 발의 목적 위치를 결정하기 위한 3가지 요소들을 식 (3), (4)를 통해 설명한다.

발의 목적위치에 구속조건을 추가하여 두 다리가 서로 충돌하지 않도록 한다. 두 다리의 충돌이 예상될 경우 식 (5)을 이용하여 이동하는 발의 목적위치를 변경하고, 다음 단계에서 회전을 통해 발이 목적위치에 자연스럽게 이동할 수 있도록 한다.

$$\begin{aligned} \varphi_d &= \left(\frac{\varphi}{\pi/2}\right)\varphi_{\max}, \tau = \left(\frac{\varphi}{\pi/2}\right)\tau_{\max}, & (5) \\ &\text{for } 0 \leq \varphi < \pi/2, w = \text{right} \\ \varphi_d &= \left(\frac{\varphi - \pi}{\pi/2}\right)\varphi_{\max} + \pi, \tau = \left(\frac{\varphi - \pi}{\pi/2}\right)\tau_{\max}, \\ &\text{for } \pi/2 \leq \varphi < \pi, w = \text{right} \\ &\text{or } \pi \leq \varphi < 3\pi/2, w = \text{left} \\ \varphi_d &= \left(\frac{\varphi - 2\pi}{\pi/2}\right)\varphi_{\max} + 2\pi, \tau = \left(\frac{\varphi - 2\pi}{\pi/2}\right)\tau_{\max}, \\ &\text{for } 3\pi/2 \leq \varphi < 2\pi, w = \text{left} \end{aligned}$$

$\varphi$ 는 캐릭터의 전면과 이동 발의 목적위치가 이루는 각도를,  $\varphi_d$ 는 수정된 발의 목적위치 각도를 나타낸다.  $w$ 은 이동 다리가 왼쪽인지 오른쪽인지를 나타낸다.  $\varphi_{\max}$ 는 발이 위치할 수 있는 최대 허용각도를 나타낸다.  $\tau_{\max}$ 는 수직축 방향으로 중심에 줄 수 있는 최대 토크를,  $\tau$ 는 실제 적용될 토크를 나타낸다. 그림 4는 외력이 인체 오른쪽 뒤에서 작용하였을 때 오른쪽 발의 목적위치를 나타내고 있다. 발의 목적 위치가 현재 위치와 멀어질수록 더 큰 회전을 통하여 목적 위치로 이동할 수 있게 된다.

#### 4.3 관절 제어

발의 목적 위치가 결정된 후에는 발을 내딛기 위해 발을 들어주는 과정이 필요하다. 고관절과 무릎관절에 토크를 적용하여 그림 5의 (b)와 같은 자세를 만들어 주는데, 관절의 현재 각도에서 원하는 각도를 만들어 주기 위해 식 (2)의 PD-controller를 사용한다. 인체를 지탱하고 있는 반대편 다리도 다리가 굽어 주저앉게 되는 것을 막기 위해 다리를 퍼주는 방향으로 고관절과 무릎관절에 토크를 적용한다. 시뮬레이션이 진행되는 동안 발목관절은 지면과 평행한 각을 목적각도로 유지한다. 이를 통해 발이 지면에 닿는 면적은 최대가 되고 캐릭터는 보다 안정적으로 균형을 유지한다. 이 단계는 정해진 시간이 경과하거나 원하는 고관절과 무릎관절의 각도를 만족하면 다음 단계로 넘어간다.

#### 4.4 동작 제어

동작 제어 단계에서는 추가적인 힘을 사용하여 발을 목적 위치에 안정적이고 자연스럽게 도달할 수 있도록 제어한다. 동작 제어는 중심의 회전, 머리를 중심축에 수직인 방향으로 세우는 힘 그리고 다리를 목적위치에 도달하도록 발에 주는 힘으로 구성된다. 발을 목적위치로 단순히 뺄 경우 반대편의 다리와 충돌이 있을 수 있는데 이 경우 몸의 회전이 필요하다. 이를 위해 식 (5)을 바탕으로 캐릭터의 중심에 지면과 수직인 방향의 토크를 주어 회전한다. 또한 캐릭터가 중력에 의해 지면에 넘어지는 것을 막고 상체의 균형 유지를 위해, 머리를 중심축에 수직인 방향의 위치로 힘을 가하여 이동시켜 준다. 이동하기 위해 든 다리는 발의 목적위치 방향으로 점진적인 힘을 적용하여 자연스럽게 발을 이동시킨다. 그림 5의 (c)는 이러한 힘들의 제어를 나타내고 있다.

발이 목적위치에 도달하면 발의 위치를 고정시켜주는 힘을 주어 발이 미끄러지는 것을 막고, 지면에 붙어 있던 반대편 발은 띄워 관절 제어 단계를 반복한다. 균형 유지의 적합성은 캐릭터의 지면과의 기울어진 각을 바탕으로 판단하고, 부적합할 경우 렉돌 시뮬레이션을 사용하여 자연스럽게 넘어지는 모션을 연출한다. 발의 목적위치 결정 단계에서 두 발이 모두 운동학적 모델의 발의 위치와 일치되도록 선정되고 발이 목적위치에 도달한 경우, 모션 합성 기법을 통해 운동학적 제어 단계로 전이한다.

#### 5. 구현 및 결과

실험에 사용된 환경은 Intel Core2 2.4 GHz 프로세서와 2GB 램(RAM) 그리고 ATI Radeon X1900 그래픽 카드가 장착된 PC를 사용하였다. Windows 플랫폼 환경에서 DirectX 9.0기반으로 구현되었다. 강제 시뮬레이션, 관절의 구성 그리고 충돌 처리를 위해 공개 물리 엔진인 Open Dynamics Engine(ODE)를 사용하였다[12]. 연산 수행 속도는 초당 평균 3441 프레임을 보였고 이는 실시간 프로그램에 사용하기에 충분한 수치이다.

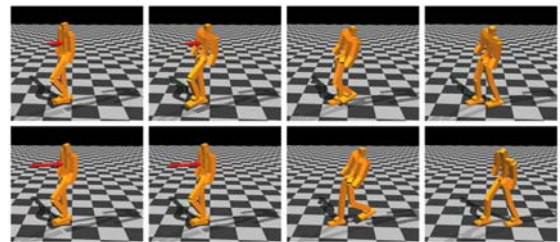


그림 6. 다른 크기의 외력에 대한 반응 모션

먼저, 그림 6은 강도가 다른 외력이 배에 가해졌을 때의 반응 모션을 나타낸다. 외력의 크기가 커질수록 균형을 잡기



위한 캐릭터 발의 보폭도 커지게 된다.

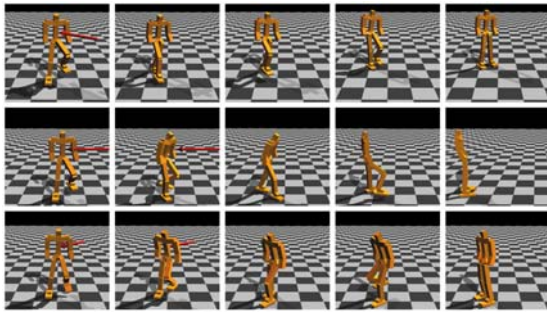


그림 7. 다른 방향의 외력에 대한 반응 모션

두 번째로 다른 방향에서 외력이 가해졌을 때의 반응 모션의 예를 나타낸다. 그림 7은 서있는 모션을 사용하였고 배의 앞과 옆 그리고 뒤에서 외력이 가해졌을 때의 반응 모션들을 나타낸다. 캐릭터 옆 방향에서 외력이 가해질 경우 식(5)에 의한 토크의 영향으로 캐릭터는 자연스럽게 회전한다.

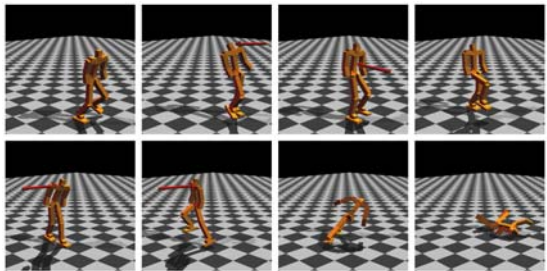


그림 8. 연속적인 외력에 대한 반응 모션

세 번째로 그림 8는 걷고 있는 모션의 머리 앞 방향으로 외력을 주고 이어서 바로 가슴과 왼쪽 어깨에 외력을 주어, 연속적인 외력에 대한 반응 모션을 보이고 있다. 캐릭터는 시뮬레이션 되는 동안 다양한 방향과 다른 크기의 외력을 연속적으로 받을 수도 있고 미는 것과 같은 지속적인 외력에 대해서도 반응 모션을 생성한다. 그림 8의 마지막 외력에 캐릭터가 넘어진 것은 캐릭터가 균형을 잃어 더 이상 반응 모션을 생성할 수 없게 되었기 때문이다.

## 6. 결론 및 향후 과제

본 논문은 사용자가 가하는 외력이 인간형 캐릭터에 적용되었을 때 반응 모션을 생성하는 기법을 제시하였다. 발의 지면과의 접촉 상태와 외력의 방향에 근거하여 움직일 발을 결정하고 발을 내딛는 몇 가지의 행동 제어를 통해 균형을 잡아주는 과정을 보였다. 본 기법은 연속적으로 가해지는 외력과 미는 것과 같은 지속적인 외력에도 자연스러운 반응 모션을 생성을 보인다. 또한 미리 캡처된 반응 모션 데이터를 활용하지 않고서도 자연스럽게도 빠른 반응 모션을 생성한다. 따라서 제안된 방법은 외력에 실시간으로 반응하는 모션이 필요한 여러 응용에 효과적으로 적용이 가능하다. 향후

과제로는 조금 더 자연스러운 모션의 생성을 위해 상하체 분리를 이용한 균형 제어 기술과의 접목이 있을 수 있겠다.

## 참고문헌

- [1] Z. Popović, A. Witkin, "Physically Based Motion Transformation," In Proceedings of SIGGRAPH 1999, pp. 11~20, 1999.
- [2] M. van de Pann, "Parameterized Gait Synthesis," Computer Graphics and Applications, Vol.16, No.2, pp. 40~49, 1996.
- [3] J. K. Hodgins, W. L. Wooten, D. C. Brogan, J. F. O'Brien, "Animating human athletics," In Proceedings of the 22nd Annual Conference on Computer Graphics and Interactive Techniques, ACM Press, pp. 71~78, 1995.
- [4] P. Faloutsos, M. van de Panne, D. Terzopoulos, "Composable controllers for physics-based character animation," Proceedings of ACM SIGGRAPH 2001, pp. 251~260, 2001.
- [5] V. B. Zordan, J. K. Hodgins, "Motion capture-driven simulations that hit and react," In Proceedings of the 2002 ACM SIGGRAPH/Eurographics symposium on Computer animation, ACM Press, pp. 89~96, 2002.
- [6] M. Oshita, A. Makinouchi, "A dynamic motion control technique for human-like articulated figures," In Proceedings of the Eurographics 2001, Vol.20, No.3, pp. 192~202, 2001.
- [7] T. Komura, E. S. L. Ho, R. W. H. Lau, "Animating Reactive Motion Using Momentum-based Inverse Kinematics and Captured Motion," Computer Animation and Virtual Worlds Journal (special issue of CASA2005), Vol.16, pp. 213~223, 2005.
- [8] K. Yin, K. Loken, M. van de Panne, "SIMBICON: Simple biped locomotion control," ACM Transactions on Graphics, Vol. 26, No. 3, 2007
- [9] V. B. Zordan, A. Majkowska, B. Chiu, M. Fast, "Dynamic response for motion capture animation," Proceedings of ACM SIGGRAPH 2005, Vol.24, No.3, pp. 697~701, 2005.
- [10] Ari Shapiro, Fred Pighin, "Hybrid Control For Interactive Character Animation," Pacific Conference on Computer Graphics and Applications 2003, pp. 455~461, 2003.
- [11] P. Faloutsos, "Composable Controllers for Physically-based Character Animation," PhD thesis, Department of Computer Science, University of Toronto, 2002.
- [12] R. Smith, Open dynamics engine (ODE): A rigid body dynamics library. <http://www.ode.org/>, 2004.