한국컴퓨터그래픽스학회
Korea Computer Graphics Society Vol. 27, No. 3, P. 13~23

심층 강화 학습을 활용한 단일 강체 캐릭터의 모션 생성

안제원[1O] 구태홍[2] 권태수[3*]
한양대학교 지능융합학과[1] 한양대학교 컴퓨터 소프트웨어학과[2,3*]
ahnjawon@naver.com[1], {gestoru[2], taesoobear[3*]}@gmail.com

Motion Generation of a Single Rigid Body Character Using Deep Reinforcement Learning

Jewon Ahn[1O] Taehong Gu[2] Taesoo Kwon[3*]
Dept. of Intelligence Convergence[1] Dept. of Computer and Software, Hanyang University[2,3*]

요 약

본 논문에서는 단일 강체 모델(single rigid body)의 무게 중심(center of mass) 좌표계와 발의 위치를 활용하여 캐릭터의 동작을 생성하는 프레임워크를 제안한다. 이 프레임워크를 사용하면 기존의 전신 동작(full body)에 대한 정보를 사용할 때 보다 입력 상태 벡터(input state)의 차원을 줄임으로써 강화 학습의 속도를 개선할 수 있다. 또한 기존의 방법보다 학습 속도를 약 2 시간(약 68% 감소) 감소시켰음에도 기존의 방법 대비 최대 7.5배(약 1500 N)의 외력을 더 견딜 수 있는 더욱 견고한(robust) 모션을 생성할 수 있다. 본 논문에서는 이를 위해 무게 중심의 다음 좌표계를 구하기 위해 중심 역학(centroidal dynamics)을 활용하였고, 이에 필요한 매개 변수(parameter)들과 다음 발의 위치와 접촉력 계산에 필요한 매개 변수들을 구하는 정책(policy)의 학습을 심층 강화 학습(deep reinforcement learning)을 사용하여 구현하였다.

Abstract

In this paper, we proposed a framework that generates the trajectory of a single rigid body based on its COM configuration and contact pose. Because we use a smaller input dimension than when we use a full body state, we can improve the learning time for reinforcement learning. Even with a 68% reduction in learning time (approximately two hours), the character trained by our network is more robust to external perturbations tolerating an external force of 1500 N which is about 7.5 times larger than the maximum magnitude from a previous approach. For this framework, we use centroidal dynamics to calculate the next configuration of the COM, and use reinforcement learning for obtaining a policy that gives us parameters for controlling the contact positions and forces.

키워드: 심층 강화 학습, 중심 역학 모델, 일체형 강체, 물리 기반 모델, 무게 중심

Keywords: deep reinforcement learning, centroidal dynamics models, single rigid body, physics–based model, center of mass

1. 서론

액체나 기구 같은 것들에 대한 물리-기반 시뮬레이션은 산업 전반에 걸쳐 흔하게 볼 수 있는 반면, 물리적으로 시뮬레이션 되는 캐릭터(character)들은 상대적으로 그렇지 않다. 특히 인간이나 동물들의 모션을 모델링(modeling)하는 것은 여전히 어려운 과제로 남아있다. 심층 강화 학습(deep reinforcement learning)은 에이전트(agent)가 시행 착오를 통해 다양한 기술을 수행하는 방법을 학습하는 기법으로, 모션 합성(motion synthesis)을 위한 좋은 모델링을 제공한다. 주로 가상 캐릭터의 전신 동작(full body)의 시뮬레이션을 활용하며, 수작업으로 컨트롤러를

*corresponding author: Taesoo Kwon/Hanyang University(taesoobear@gmail.com)

Received : 2021.06.09./ Review completed : 1st 2021.06.18. / Accepted : 2021.06.25.
DOI : 10.15701/kcgs.2021.27.3.13
ISSN : 1975-7883(Print)/2383-529X(Online)

디자인하는 것 보다 인간의 통찰력에 대한 필요성을 줄여주었다. 그러나 전신 동작 시뮬레이션을 통해 모션 합성을 위한 학습을 진행하는 경우, 상태(state) 공간과 액션(action) 공간의 차원이 크고, 학습 속도가 느린 단점이 있다. 그러므로 차원을 줄인 상태(reduced state)에 대한 심층 강화 학습 모델을 모델링하는 것이 요구된다.

본 논문에서는 단일 강체 모델(single rigid body)의 무게중심(center of mass)에 적용되는 선형 힘(linear force), 토크(torque), 그리고 발의 위치(contact position)를 활용하여 단일 강체 모델의 동작을 생성하고, 심층 강화 학습을 활용하여 정책(policy) 네트워크를 얻을 수 있는 프레임워크를 제안한다. 캐릭터를 단일 강체로 근사하기 때문에 전신 동작을 사용했을 때 보다 강화 학습의 속도를 개선할 수 있다. 또한 우리는 제안한 단일 강체 모델의 동작을 생성하는 프레임워크가 전신 모델을 활용하는 기존 연구[1]보다 외력에 더 견고하다는(robust) 것을 실험적으로 보였다.

즉, 본 논문에서 제안한 단일 강체 모델의 동작을 생성할 수 있는 프레임워크의 장점은 다음과 같다.

1. 상태(state)와 액션(action) 벡터의 차원을 감소시켜 기존의 강화 학습 속도를 개선하였다.

2. 기존의 강화 학습[1]과 비교하여 더 나은 견고함(robustness)을 가진다.

3. 중심 역학(centroidal dynamics)을 활용하는 전신 동작의 모션 합성에서 기존의 경로 최적화(trajectory optimization)를 대체하는 프레임워크를 제공한다.

2. 관련 연구

캐릭터 모션은 여러 방식으로 합성될 수 있다. 우리는 이와 관련된 기전 연구들 중 본 논문과 직접적으로 관련이 있는 물리–기반 모델(physics–based models), 심층 강화 학습(deep reinforcement learning), 그리고 중심 역학 모델(centroidal dynamics models)과 관련된 연구에 집중한다.

2.1 물리–기반 모델

시뮬레이션 되는 캐릭터에 대한 컨트롤러를 설계하는 것은 어려운 문제로 남아있다. 이 가운데 인간과 비인간 캐릭터를 대상으로 발전해온 견고한(robust) 컨트롤러들과 보행동작(locomotion)은 많은 연구가 진행되었다 [2, 3, 4]. 이런 컨트롤러들은 단순화된 모델과 최적화 과정을 통해 얻어지며, 원하는 동작을 달성하기 위해 매개 변수(parameter)들의 집합이 조정된다 [5, 6, 7]. 또한 이동을 위한 컨트롤러들을 발전시키기 위해 이차 계획법(quadratic programming)을 활용하는 동역학 최적화 방법들도 적용되었다 [8, 9, 10]. 경로 최적화 또한 연구되어 왔는데, 이는 다양한 과제들과 캐릭터들을 위한 물리적으로 설득력 있는 모션들을 합성하기 위해 진행되었다 [11, 12]. 이러한 방법들은 운동 방정식들을 제약조건으로 설정하며 모션들을 합성하는데, 이는 오프라인 최적화 과정을 통해 이루어진다. 최근 연구에는 이러한 기술들을 온라인 모델-예측(online model-predictive) 제어 방법들로 발전시켜왔다 [13, 14]. 그러나 이는 모션의 품질과 장기적인 계획(long-term planning) 능력 측면에서 한계점을 갖고 있다.

2.2 심층 강화 학습

시뮬레이션 되는 캐릭터들을 제어하는 컨트롤러들을 발전시키 위해 사용되는 많은 종류의 최적화 기술들이 강화 학습(reinforcement learning)을 기초로 활용하고 있다. 가치 반복(value iteration) 방법들은 모션 클립들을 순서대로 배열하기 위한 운동학(kinematics)을 사용한 컨트롤러들을 발전시키기 위해 사용되어 왔다 [15, 16]. 또한 이와 비슷한 방법들도 연구되었다 [17, 18]. 이러한 강화 학습에 최근 심층 신경망(deep neural networks)이 적용되며 더 다양한 도전적인 과제들을 수행할 수 있게 되었다 [19, 20, 21, 22, 23, 24]. 하지만 몇 개의 도전들은 자연적인 움직임을 위한 보상 함수(reward function)들을 확립하는 것의 어려움과 자연스럽게 시뮬레이션 되는 보행동작(locomotion)을 달성하기 위해 사용되는 생체역학 모델(biomechanical models)들의 부재로 수행이 잘 이루어지지 않았으며, 이에 대한 연구도 진행되었다 [7, 10].

2.3 중심 역학 모델

중심 역학 모델(centroidal dynamics models)은 경로에 대한 최적화 문제를 크게 단순화시킬 수 있는 장점을 갖고 있다. 중심 역학 모델은 모션을 무게 중심을 중심으로 여기에 작용되는 전체 선형 운동량(linear momentum), 각 운동량(angular momentum) 그리고 이에 해당하는 바디의 방향으로 모델링을 진행한다. 즉, 선형 및 각 운동량, 각 속도(angular velocity), 선 속도(linear velocity)가 각 3차원을 나타내므로 총 12차원에 해당하는 변수들이 현재 발의 위치와 힘들에 대한 정보들과 결합되어 시간에 따라 발전되며 가장 역동적인 다리의 움직임에 대한 주요 부분들을 모델링 한다. 이러한 중심 역학 모델의 이점은

결국 휴머노이드(humanoid) 모션들에 잘 적용되었다 [25]. 본 논문 또한 이러한 중심 역학 모델을 활용하여 단일 강체 모델의 모션을 레퍼런스(reference)에 맞게 생성하였다. 최근에는 실현 가능한 중심 역학 및 관절 각도(joint angles) 그리고 충돌 제약 조건(constraints)에 대한 일부 기능들을 통합하기 위해 단일 솔루션이 제안되었다 [26]. 또한 중심 역학 모델의 경로 최적화는 ANYmal 로봇으로의 이동에 대한 설명을 포함하여 외족 보행 로봇(monopods), 이족 보행 로봇(bipeds) 그리고 사족 보행 로봇(quadrupeds)에 대해 표현되며 중심 역학 모델에 대한 다른 연구들도 진행되었다 [27, 28, 29].

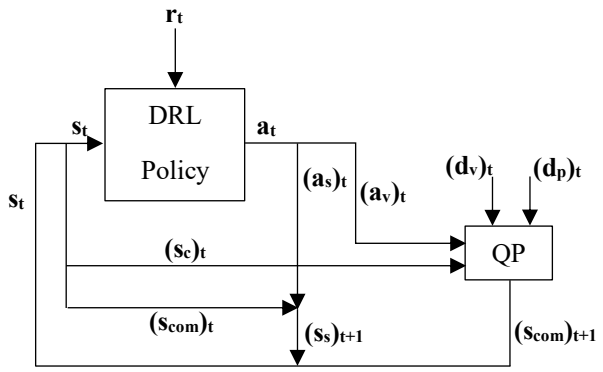


Figure 1: System Overview

3. 개요

본 논문에서는 먼저 중심 역학을 적용하기 위한 단일 강체 모델을 준비하였다. 단일 강체(single rigid body) 캐릭터의 발의 접촉 타이밍(contact timing)을 레퍼런스(reference) 캐릭터의 타이밍에 맞추어 발의 접촉 타이밍을 정의하였고, 단일 강체 캐릭터가 이에 맞추어 지면과의 접촉이 이루어지게 제어하였다. 또한 레퍼런스 캐릭터의 모션 데이터를 통해 이 후 이상 가속도(desired acceleration)를 구하기 위한 타겟이 되어 줄 이상 위치 및 속도(desired pose, desired velocity)를 준비하였다.

Figure 1은 우리가 제안한 모델의 개요이다. 이 모델은 크게 고수준 컨트롤러(high-level controller)인 심층 강화학습(deep reinforcement learning)을 통해 학습되는 정책(policy)과 저수준 컨트롤러(low-level controller)인 이차 계획법(quadratic programming (QP))으로 구성된다. 스텝(step) t가 진행되면 먼저 정책으로 상태(state) 벡터($\mathbf{s}_t$)가 입력된다. 상태 벡터($\mathbf{s}_t$)는 현 프레임에 대한 단일 강체 캐릭터의 무게 중심 정보($(\mathbf{s}_{com})_t$)와 두 발에 대한 정보, 그리고 현 단계(phase)를 나타내는 단계 변수(phase variable)로 이루어져 있다. 상태 벡터가 입력되면 정책으로부터 액션(action) 벡터를 얻게 된다. 액션 벡터는 단일 강체 캐릭터의 무게 중심의 속도 정보($(\mathbf{a}_v)_t$)와 업데이트 될 두 발의 위치 정보를 포함한다. 상태, 액션 벡터에 관한 내용은 상태와 액션(section 5.1)에서 자세히 다루겠다. 정책을 통해 액션이 얻어지면 액션 벡터 성분 중 단일 강체 캐릭터의 무게 중심의 속도에 관한 성분($(\mathbf{a}_v)_t$)은 데이터 전 처리 과정(data preprocessing)을 통해 얻은 레퍼런스 캐릭터 모션의 이상 속도 및 위치(desired velocity & position, $(\mathbf{d}_v)_t$, $(\mathbf{d}_p)_t$), 그리고 위 현 스텝 t의 상태 벡터 성분 중 단일 강체 캐릭터의 지면과 접촉된 발의 위치 정보($(\mathbf{s}_c)_t$)와 함께 이차 계획법의 입력 벡터가 된다. 즉, 액션의 단일 강체 캐릭터의 무게 중심의 속도 정보($(\mathbf{a}_v)_t$)는 레퍼런스 캐릭터 모션 좌표계의 이상 속도($(\mathbf{d}_v)_t$)와 함께 이차 계획법에서 이용되는 이상 속도를 구체화하는데 사용된다. 이차 계획법 입력 벡터를 활용하여 이차 계획법은 다음 프레임(t+1)의 단일 강체 캐릭터의 무게 중심 좌표계를 계산하며 이는 다음 프레임(t+1)을 위한 상태 벡터($\mathbf{s}_{t+1}$) 성분이 된다. 이차 계획법을 통한 단일 강체 캐릭터의 무게 중심 좌표계의 계산에 대한 구체적인 설명은 추적(tracking) (section 7)에서 진행하겠다. 마지막으로 다음 상태 벡터에 포함될 스윙(swing) 단계의 단일 강체 캐릭터의 발에 대한 위치 정보($(\mathbf{s}_s)_{t+1}$)는 위에서 얻어진 액션 벡터 성분들 중 단일 강체 캐릭터의 발의 위치 정보($(\mathbf{a}_s)_t$)와 이 전 프레임에서 계산된 단일 강체 캐릭터의 무게 중심 좌표계, 즉, 현 스텝 t의 상태 벡터 성분들 중 단일 강체 캐릭터의 무게 중심 정보($(\mathbf{s}_{com})_t$)를 통해 계산된다. 계산된 스윙 상태의 단일 강체 캐릭터의 발에 대한 정보($(\mathbf{s}_s)_{t+1}$)는 다음 프레임을 위한 상태 벡터($(\mathbf{s}_{t+1})$) 성분에 포함된다. 단일 강체 캐릭터의 발의 위치 계산에 대해서는 접촉점 계산(section 6)에서 자세히 설명하겠다. 구해진 정보들을 토대로 현 스텝 t에 대한 보상(reward, $\mathbf{r}_t$) 값이 계산되며, 정책의 학습에 고려된다. 정책은 심층 신경망(deep neural networks)을 사용하였으며, proximal policy optimization algorithm (PPO)를 사용하여 학습을 진행하였다 [30].

4. 데이터 준비

본 논문에서는 학습에 앞서 레퍼런스 캐릭터의 모션으로부터 데이터 전 처리(data preprocessing)가 진행되어야 한다. 이차 계획법 목적 함수(objective function)에 사용될 이상 가속도를 계산하기 위해 이상 위치 및 속도를 레퍼런스 캐릭터의 모션을 통해 먼저 구해야 한다. 또한 단일 강체 캐릭터의 발이 지면과 접촉하는 시점을 제어

하는데 활용되는 단일 강체 캐릭터의 발의 접촉 타이밍을 레퍼런스 캐릭터의 모션을 통해 미리 지정하였다.

이상 위치는 레퍼런스 캐릭터의 모션 데이터 중 중심 역학을 적용하기 위해 무게 중심에 해당되는 골반의 좌표계를 각 프레임 별로 가져와 저장하였다. 골반의 좌표계는 글로벌 이동(global translation(x, y, z)), 그리고 쿼터니언(quaternion(w, x, y, z)) 데이터로 구성된다. 이상 속도 또한 레퍼런스 캐릭터의 무게 중심 좌표계의 t, t +1 값을 통해 각 프레임 별로 얻어지고 저장된다. 발이 지면과 접촉하는 오른발의 접촉 타이밍은 35 프레임으로, 왼발의 경우에는 17 프레임으로 정하였다. 발이 스윙 단계(swing phase)로 진입하는 발을 지면으로부터 떼는(touch off) 타이밍도 정의되었다. 오른발을 지면으로부터 떼는 타이밍은 19 프레임으로 정했고, 왼발의 경우는 37 프레임으로 정하였다. 만약 왼발의 스윙이 이루어질 때 53 프레임에 도달하면 왼발을 지면으로부터 떼는 타이밍이 17 프레임에서 잘 이루어지게 하기 위해 53 프레임에서 16 프레임으로 이동시키는 레퍼런스 캐릭터의 프레임 제어를 진행하였다.

단일 강체 캐릭터는 무게 중심이 정의된 박스(box) 형태로 생성하였다(Figure 2 참조). 단일 강체 캐릭터의 무게 중심 좌표계는 글로벌 이동(global translation(x, y, z)), 그리고 쿼터니언(quaternion(w, x, y, z)) 데이터로 구성되며, 발의 위치 및 접촉점은 Figure 2를 통해 알 수 있다. 또한 질량은 60kg으로 설정하였다.

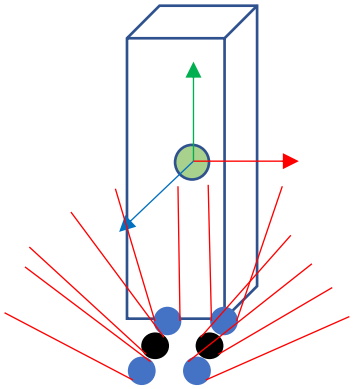


Figure 2: The structure of single rigid body (SRB) character. The black circles indicate the left and right feet locations which are used for calculating 4 contact points indicated by blue circles. The red lines represent friction cones that constrain contact forces. The green circle represents the position and orientation of a frame located at the center of mass (COM) of the SRB character.

5. 정책 표현(Policy representation)

레퍼런스 캐릭터의 모션이 주어질 때 정책(policy)의 목적은 주어진 물리 환경 아래에서 레퍼런스 캐릭터의 모션에 비슷한 단일 강체 캐릭터의 모션을 만들 수 있는 적합한 액션을 각 스텝(step)별로 결정하는 것이다. End-to-end 심층 강화 학습을 위한 정책 네트워크 π는 신경망으로 이루어져 있으며, 상태 s가 주어지면 이를 액션과 연결(mapping)하여 준다.

정책 네트워크(policy network)는 네 개의 완전히 연결된 층(fully connected layer)으로 이루어져 있으며, 각 숨겨진 층(hidden layer)은 64개의 유닛들(units)로 구성되며, 숨겨진 층의 유닛들(hidden units)에는 활성화 함수(activation function)로 tanh가 사용된다. 또한 가치 함수 네트워크(value function network)도 같은 네트워크 구조를 가지는데, 한 가지 차이점은 출력(output)이 정책 네트워크는 액션의 차원만큼 선형 출력 층(linear output layer)을 갖는 반면 가치 함수 네트워크는 하나의 선형 유닛(linear unit)을 가지는 것이다. Figure 3은 위에서 언급한 정책 네트워크의 도식적인 그림을 나타낸다.

5.1 상태와 액션

상태(state)는 총 20차원으로 크게 단일 강체 캐릭터의 무게 중심 관련 상태가 11차원, 단일 강체 캐릭터의 발의 위치에 관하여 8차원, 마지막으로 단일 강체 캐릭터의 현 상태를 나타내어주는 단계 변수(phase variable)로 구성된다. 0~4 인덱스(index)에는 t+1에서의 단일 강체 캐릭터의 무게 중심 좌표계가 입력되는데, 순서대로 단일 강체 캐릭터의 무게 중심의 높이(y), t+1에서의 y축 회전 값 즉, 정면 방향(forward direction)에 대한 로컬 쿼터니언 데이터(local quaternion data)가 적용된다. 5~10 인덱스에는 마찬가지로 t+1에서의 단일 강체 캐릭터의 무게 중심의 정면 방향에 대한 로컬 좌표계(local coordinate)에서 정의된 선 속도, 각 속도가 포함된다. 11~13 인덱스에는 t에서의 단일 강체 캐릭터의 좌표계에 대한 오른발의 로컬 위치(x, y, z)가 입력되며, 14 인덱스에는 만약 오른발의 단계가 스윙 즉, 지면에 고정된 것이 아닌 움직이고 있는 상태라면 t+1 일 때의 단일 강체 캐릭터의 무게 중심 정면 방향과 t 일 때의 오른발의 정면 방향 차이를 넣어준다. 현 상태에서의 액션에 대한 상태가 무게 중심은 t+1에 해당하는 값이며, 발의 위치는 t에 해당하는 값이므로 그 둘의 차이가 들어가게 된다. 만약 단일 강체 캐릭터의 오른발의 단계가 스윙 단계가 아니라면 즉, 지면에 고정된 단계라면 '0'이 들어간다. 15~18 인덱스에는 11~14와 동일한 내용이 왼발에 대하여 적용된다. 19 인덱스에는 단일 강체 캐릭터의 단계를 나타내는 단계 변

수 $\phi \in [0, 0.17]$가 입력된다. 우리가 고안한 프레임워크는 발의 접촉 타이밍을 미리 정해주므로 이 단계 변수는 현 단계를 제시해 준다. $\phi = 0$은 단일 강체 캐릭터의 모션의 시작을 나타내며, $\phi = 0.17$은 왼발의 접촉 타이밍 조절에 들어가기 전 단계를 나타낸다.

액션은 총 10차원으로 크게 단일 강체 캐릭터의 무게 중심의 이상 상대 선 속도 및 각 속도(desired relative linear & angular velocity), 그리고 단일 강체 캐릭터의 좌표계(SRB body frame)에 대한 발의 로컬 위치(local contact pose)로 구성된다. 먼저 0~3 인덱스에는 오른발, 왼발의 단일 강체 캐릭터의 좌표계에 대한 발의 로컬 위치(x, z)가 포함된다. 이 후 4~6 인덱스에는 레퍼런스 캐릭터의 좌표계(reference body frame)에 대한 단일 강체 캐릭터의 무게 중심의 이상 상대 선 속도가 들어가며, 7~9 인덱스에는 레퍼런스 캐릭터의 좌표계에 대한 이상 상대 각 속도가 입력된다. 액션의 속도 정보들은 이차 계획법 컨트롤러를 위한 이상 가속도를 구체화하는데 사용된다.

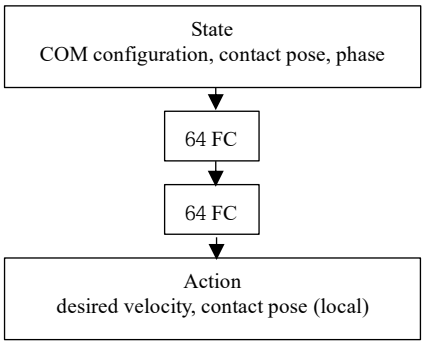


Figure 3: Schematic illustration of the policy network.

5.2 보상

각 스텝 t에서의 보상(reward) r_t는 레퍼런스 캐릭터의 모션을 단일 강체 캐릭터가 제대로 따라갈 수 있게 제어해 준다.

$$r_t = w^s r_t^s - w^m r_t^m$$

r_t^s는 단일 강체 캐릭터가 똑바로 서 있는 것을 고려한 부분(stand up reward term)으로 정해진 조건 내에 단일 강체 캐릭터가 존재한다면 1을 부여한다. 이 조건에 대해서는 이 후 종료 조건(section 5.3)에서 다루겠다. 만약 조건을 벗어난다면 0을 부여하고 에피소드(episode)를 종료한다. 이 경우 전체 보상 값이 음의 값을 가질 수 있는데 이때의 보상은 0으로 처리하였다. 이 보상 부분(reward term)에 대한 가중치(weight) w^s은 5로 정하였다. r_t^m는 단일 강체 캐릭터가 레퍼런스 캐릭터를 잘 따라갈 수 있게 제어하는 모방 보상 부분(mimic reward term)이다. 이 보상 부분에 대한 가중치 w^m는 0.3으로 정하였다.

$$r_t^m = w^p r_t^p + w^e r_t^e + w^v r_t^v$$
$$w^p = 16.25, w^e = 2.03, w^v = 0.1$$

r_t^p는 t와 t+1사이의 단일 강체 캐릭터의 무게 중심의 각도 및 선형 변화량과 레퍼런스 캐릭터의 무게 중심의 각도 및 선형 변화량을 고려한 포즈 보상(pose reward) 부분이다. 단일 강체 캐릭터의 무게 중심이 다음 프레임으로 이동할 때 레퍼런스 캐릭터의 무게 중심과 비슷한 위치에 존재할 수 있게 제어한다. $(d_{SRB})_t$, $(d_{ref})_t$는 각각 단일 강체, 레퍼런스 캐릭터에 대해 t일 때 무게 중심의 정면 방향에 대한 t, t+1사이의 무게 중심 선형 변화량을 가리킨다. $(q_{SRB})_t$, $(q_{ref})_t$는 각각 단일 강체, 레퍼런스 캐릭터에 대해 t일 때의 무게 중심의 y축 회전 쿼터니언과 t+1의 쿼터니언 사이의 변화량(angular)을 나타낸다. 스텝 t일 때의 무게 중심 y축 쿼터니언 즉, 정면 방향을 기준으로 하였으므로 다음 프레임에서 정면 방향이 얼마나 돌았고, 바디(body)가 얼마나 기울었는지 고려할 수 있다. $q_s \ominus q_r$는 쿼터니언 차이를 나타내며, $\|q\|$는 두 좌표계 사이의 최소 회전각을 의미한다.

$$r_t^p = \left\| (d_{SRB})_t - (d_{ref})_t \right\|_2 + 5 \left\| (q_{SRB})_t \ominus (q_{ref})_t \right\|$$

r_t^e는 두 발의 위치에 관련된 엔드 이펙터(end-effector) 보상으로서, 단일 강체 캐릭터의 스윙 단계인 발과 스윙 단계가 아닌 발의 단일 강체 캐릭터의 좌표계에 대한 로컬 위치가 레퍼런스 캐릭터의 좌표계에 대한 해당 로컬 위치와 비슷하게 제어해준다. 현재 지면과 접촉된 발의 위치는 다음 스텝(t+1)의 무게 중심 위치에 영향을 미치므로 현 스텝(t)의 단일 강체 캐릭터의 좌표계에 대한 다음 스텝(t+1)의 무게 중심의 로컬 위치도 레퍼런스 캐릭터를 따라갈 수 있게 제어하였다. $(p_{SRB})_t^j$, $(p_{ref})_t^j$는 각각 t일 때의 단일 강체 및 레퍼런스 캐릭터의 좌표계에 대한 j번째 접촉 포인트의 로컬 위치다. 스윙 단계에서 발생하는 힘은 지면에 접촉된 발에서의 접촉력(contact force)에서 고려되고 이상 렌더링 위치(desired rendering position)는 스윙 단계가 아닌 발에 대해 적용된다. 즉, 이차 계획법에서 다음 무게 중심 좌표계를 구하기 위해 사용되는 발은 지면과 접촉된 발이다. 그러므로 레퍼런스 캐릭터를 따라가게 제어하기 위해 사용하는 j번째 접촉 포인트는 접촉 단계의 발에 대해서만 생각한다. $(C_{SRB})_{t+1}$, $(C_{ref})_{t+1}$는 각각 t일 때의 단일 강체 및 레퍼런스 캐릭터의 좌표계에 대한 t+1에서의 단일 강체 및

레퍼런스 캐릭터의 무게 중심의 로컬 위치다.

$$r_t^e = \sum_j \left\| (p_{SRB})_t^j - (p_{ref})_t^j \right\|_2 + 4 \left\| (C_{SRB})_{t+1} - (C_{ref})_{t+1} \right\|_2$$

r_t^v는 t와 t+1 사이에서의 무게 중심 속도가 단일 강체 캐릭터와 레퍼런스 캐릭터가 비슷하도록 제어해 주는 속도 보상(velocity reward)이다. 이는 앞서 제어한 레퍼런스 캐릭터의 무게 중심과 비슷한 위치에 단일 강체 캐릭터의 무게 중심이 레퍼런스 캐릭터처럼 자연스럽게 이동할 수 있게 단일 강체의 속도를 제어한다. 즉, 현재 프레임(t)과 다음 프레임(t+1)이 서로 연관되어 고려될 수 있게 한다. $(v_{SRB})_t, (v_{ref})_t$는 t+1일 때 무게 중심의 정면 방향에 대한 t, t+1 사이에서의 무게 중심 위치 변화량에 프레임 변화율(framerate ($30\frac{1}{s}$))이 적용된 단일 강체 및 레퍼런스 캐릭터의 무게 중심 속도이다.

$$r_t^v = \left\| (v_{SRB})_t - (v_{ref})_t \right\|_2$$

5.3 종료 조건

효율적인 학습을 위해 각 에피소드 별로 정해진 구간 동안 학습이 진행될 때, 특정 조건에 따라 에피소드를 즉시 종료하고 다음 에피소드로 넘어가야 한다.

조건의 기준은 단일 강체 캐릭터의 발과 지면과의 접촉이 적절하게 이루어지는지, 그리고 단일 강체 캐릭터가 쓰러지지 않고 자세를 잘 유지하는 지의 여부로 정의하였다. 먼저 지면과 발의 적절한 접촉이 이루어지는 것을 확인하기 위해 단일 강체 캐릭터의 무게 중심의 높이, 즉 y값이 정해진 범위 내에 존재하는지 고려하였다. 범위의 최대 값은 2.0으로 설정하였고, 최소 값은 $\min(0.7(y_{ref})_{t+1}, 0.2)$ 을 통해 결정하였다. $(y_{ref})_{t+1}$는 t+1일 때의 레퍼런스 캐릭터의 y값이다. 단일 강체 캐릭터가 넘어지는 것의 기준은 무게 중심의 회전 값이 rad(70)을 넘는지에 대한 것으로 판단하였다. 또한 학습 동안 전체 스텝이 90 스텝을 넘으면 종료(termination)를 진행하였다.

6. 접촉점 계산

매 프레임 별로 각 단일 강체 캐릭터의 발 마다 2개씩 할당된 접촉 포인트들에 대한 위치의 연산은 스윙과 스윙이 아닌 단계로 나뉘어 계산된다. 각 발 마다 스윙 단계는 정해진 타이밍에 따라 정해지게 된다.

스윙 단계라면 단일 강체 캐릭터의 발의 위치는 이전 스텝(t-1)에서 계산된 단일 강체 캐릭터의 무게 중심의 위치와 현 스텝(t)에서의 액션을 활용하여 계산된다. a_t는 t일 때 단일 강체 캐릭터의 무게 중심의 정면 방향에 대한 로컬 값이므로 글로벌 값을 얻기 위해 무게 중심의 y축 회전 쿼터니언 $(q_y)_t$을 적용해준다. COM_t은 t일 때 단일 강체 캐릭터의 무게 중심의 글로벌 위치(global position)다. 오프셋(offset) k는 x축 제어를 위해 오른발의 경우 (-0.07,0,0), 왼발의 경우 (0.07,0,0)로 정의했다.

$$(p_{swing})_t = COM_t + (q_y)_t a_t + k$$

스윙 상태가 아니라면 t-1의 단일 강체 캐릭터의 발의 위치가 그대로 적용된다. 계산된 단일 강체 캐릭터의 발의 위치는 각 발 마다 접촉 포인트들의 중점이 된다. 각 발 마다 접촉 포인트에 대한 위치를 구하기 위해 계산된 발의 위치를 기준으로 z축에 대한 오프셋을 적용했다. 즉, $j \in \{1,2\}$ 번째 발에 대한 $k \in \{1,2\}$ 번째 접촉 포인트의 위치는 다음과 같다.

$$(p_k)_t = (p_j)_t + (-1)^{k-1}(0,0,0.12)$$

6.1 접촉 타이밍 제어

스윙의 경우 정해진 타이밍에 맞춰 지면과 발의 접촉이 이루어져야 한다. 그러므로 정책을 통해 단일 강체 캐릭터 모션이 업데이트 되면서 정확한 접촉 타이밍을 얻기 위해 발의 빠른 접촉(early touch down) 타이밍 및 늦은 접촉(late touch down) 타이밍에 대한 조건을 걸어 레퍼런스 캐릭터 모션의 프레임을 제어하였다.

정해진 타이밍에 따라 지면과 발의 접촉이 이루어져야 하는데 레퍼런스 캐릭터의 무게 중심의 y값이 단일 강체 캐릭터의 무게 중심의 y값 보다 작다면 아직 단일 강체 캐릭터의 발과 지면의 접촉이 잘 이루어 지지 않은 것이므로 이를 늦은 접촉 타이밍으로 고려하였다. 이 때는 레퍼런스 캐릭터 모션의 진행에 지연(delay)을 주어 단일 강체 캐릭터와의 접촉 타이밍을 맞춰주었다.

단일 강체 캐릭터의 무게 중심의 y 값이 레퍼런스 캐릭터의 접촉 순간의 무게 중심 y값인 0.9보다 작을 경우 정해진 타이밍 보다 먼저 접촉이 일어난 것이므로 이는 빠른 접촉 타이밍으로 고려하였다. 빠른 접촉 타이밍은 단일 강체 캐릭터의 발이 지면과 접촉되는 방향으로 진행 방향이 기우는 문제가 있다. 그러므로 빠른 접촉 타이밍이 발생하면 현 상태에서 가장 가까운 접촉 타이밍을 찾아 그 타이밍으로 레퍼런스 캐릭터 모션 프레임을 제어하고 단일 강체 캐릭터의 무게 중심의 y값을 '0'로

설정하였다.

7. 추적(Tracking)

본 논문에서는 단일 강체 캐릭터를 다루기 때문에 각 조인트(joints)에 적용되는 토크를 통해 다음 위치를 제어하는 비례-미분(proportional-derivative) 컨트롤러 대신 무게 중심에 걸리는 힘들을 고려하는데 용이한 이차 계획법 컨트롤러를 사용하였다.

액션과 레퍼런스 캐릭터 모션으로부터 이상 속도 및 위치가 주어지면 정의된 목적 함수와 구속 조건(constraints)을 통해 단일 강체 캐릭터의 무게 중심의 가속도가 구해진다. 계산된 가속도를 통해 단일 강체 캐릭터의 무게 중심에 적용될 접촉력(contact force)과 토크(torque)가 계산된다. 접촉력과 토크, 그리고 현재 발의 위치를 활용하여 다음 무게 중심의 좌표계가 도출된다 [31, 32].

즉, 다음 단일 강체 캐릭터의 무게 중심의 좌표계를 구하기 위한 접촉력과 토크를 계산하고자 운동 방정식(motion equation) 구속 조건과 선형화 된 접촉력의 기저(basis)를 모델링하기 위한 구속 조건을 만족시키면서 목적 함수 Q를 최소화하는 가속도를 구하였다 [33, 34]. $\ddot{\mathbf{q}}$는 단일 강체 캐릭터의 무게 중심의 가속도, $\boldsymbol{\lambda}$는 접촉력, $\boldsymbol{\tau}$는 토크, $\mathbf{J_c}$는 속도에 관련된 자코비안 행렬(Jacobian matrices), $\mathbf{B}$는 접촉력의 기저 벡터, $\mathbf{a_c}$는 무게 중심의 글로벌 가속도, 마지막으로 $\mathbf{o_c}$는 속도에 의존적인 무게 중심에서의 오프셋들을 나타낸다.

$$\min_{\ddot{\mathbf{q}}} \mathrm{Q}(\ddot{\mathbf{q}})$$

일 때

$$\mathbf{M}\ddot{\mathbf{q}} + \mathbf{b} = \boldsymbol{\tau} + \mathbf{J_c^T}\mathbf{B}\boldsymbol{\lambda},$$

$$\boldsymbol{\lambda} \geq \mathbf{0},$$

$$\mathbf{a_c} = \mathbf{B^T}\mathbf{J_c}\ddot{\mathbf{q}} + \mathbf{B^T}\dot{\mathbf{J}}_\mathbf{c}\dot{\mathbf{q}} + \dot{\mathbf{B}}^\mathbf{T}\mathbf{J_c}\dot{\mathbf{q}} \geq \mathbf{o_c}.$$

목적 함수 Q는 아래와 같이 정의하였고, $\ddot{\mathrm{q}}$는 단일 강체 캐릭터의 무게 중심의 가속도, $\ddot{\mathrm{q}}_\mathrm{d}$는 이상 가속도(desired acceleration)이다. $\mathbf{W}_\mathrm{t}$는 가중치에 대한 대각 행렬(diagonal weighting matrix)이며 무게 중심에 대한 값은 1^{-6}, 나머지 자유도들(DOFs)에 대해서는 1로 설정하였다.

$$\mathrm{Q} = (\ddot{\mathrm{q}} - \ddot{\mathrm{q}}_\mathrm{d})^\mathrm{T}\mathbf{W}_\mathrm{t}(\ddot{\mathrm{q}} - \ddot{\mathrm{q}}_\mathrm{d})$$

이상 가속도는 다음의 순서로 도출된다.

액션 a_t는 레퍼런스 캐릭터에 대한 단일 강체 캐릭터의 무게 중심의 이상 상대 선형 및 각 속도(desired relative linear & angular velocity)를 나타낸다. 단일 강체 캐릭터의 무게 중심의 이상 속도는 데이터 준비(section 4)에서 설명한 레퍼런스 캐릭터의 무게 중심의 이상 선형 및 각 속도와 액션의 합을 통해 구한다.

$$\dot{q}_d = (\dot{q}_d)_{ref} + a_t$$

액션을 통해 이상 선형 및 각 속도를 얻으면 데이터 준비(section 4)에서 설명한 레퍼런스 캐릭터의 무게 중심의 이상 위치를 활용하여 이상 가속도를 아래 정의된 식을 통해 구한다. 매개 변수 min은 -400, max는 400, a는 120, b는 35로 설정하였다.

$$\ddot{q}_d = \mathrm{clamp}(a(q_d - q) + b(\dot{q}_d - \dot{q}), min, max)$$

8. 구현

본 연구는 64GB RAM, Intel® Core™ i9-9900K CPU (16 Cores) @ 3.60GHz, GeForce RTX 2080 Ti 등으로 구성된 PC에서 진행되었고, Python ver.3.8환경에서 Pytorch ver.1.7.1을 사용하여 이차 계획법의 구현, 그리고 정책 네트워크, 가치 함수 네트워크의 구현 및 학습을 수행하였다.

정책은 m = 1024 만큼 샘플링(sampling) 되면 업데이트가 진행되며 이때 미니 배치 사이즈(minibatch size)인 n = 256 만큼 각 기울기 스텝(gradient step) 별로 샘플링 된다. 할인 계수(Discount factor) $\gamma = 0.995$로 적용하였다. 또한 TD(λ), GAE(λ)를 위해 λ = 0.95로 설정하였다. Clipped surrogate loss를 위한 우도 비율 클리핑 임계값(likelihood ratio clipping threshold)은 ϵ = 0.2, 학습률(learning rate)은 10^{-4}로 지정하였다. 학습(training)을 위해 총 2×10^8만큼의 샘플링을 진행하였으며, 학습시간은 57분 소요되었다. Figure 4는 학습된 정책을 통해 생성된 단일 강체 캐릭터의 모션에 대한 스냅 샷(snapshot)이다.

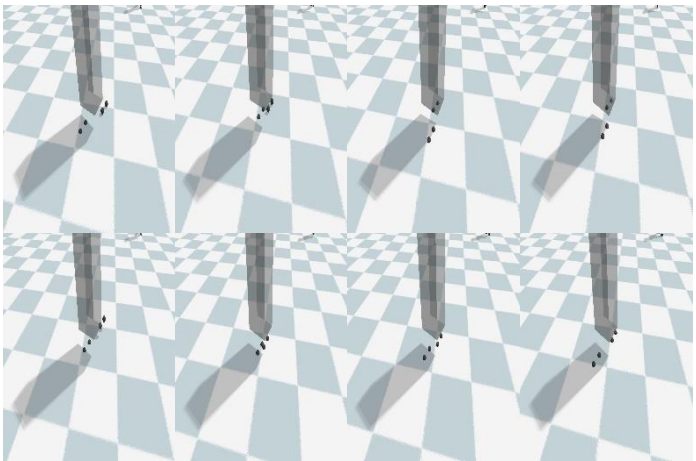
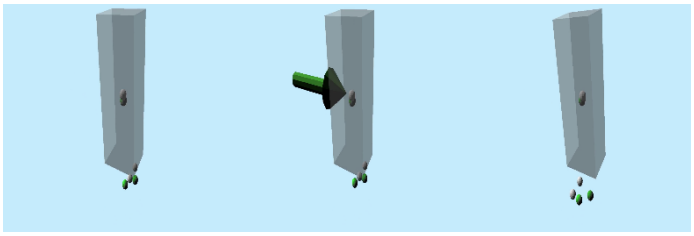


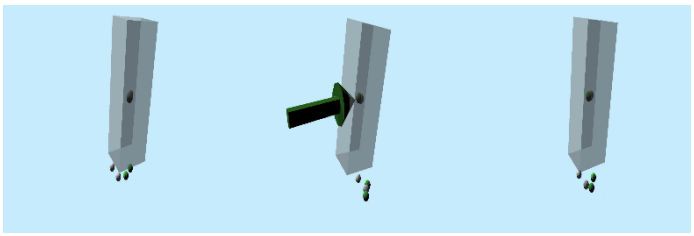
Figure 4: Snapshots of SRB motions from the trained policy. The calculated SRB contact pose and COM's position are appropriate comparing to reference motion.

9. 평가

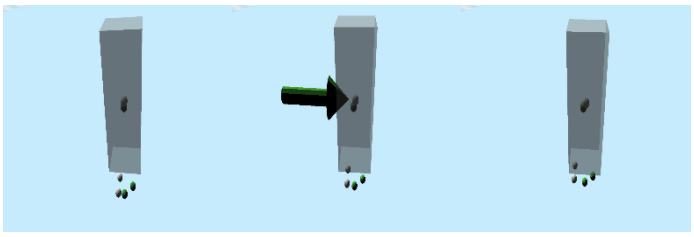
본 연구는 레퍼런스 캐릭터의 모션을 모방할 때 전신 동작에 대해서가 아닌 단일 강체 모델로 해당 모션을 모방하였다. 캐릭터를 단일 강체로 근사하기 때문에 학습속도가 전신 모델을 활용하는 기존 연구[1]를 적용한 것보다 개선되었다는 것을 알 수 있다. 우리가 제안한 정책 네트워크는 학습 때 보상(reward) 값이 수렴까지 57분이 걸린 반면, 기존 연구[1]는 학습 때 수렴까지 3시간이 걸린다. 레퍼런스 캐릭터의 무게 중심에 집중하여 학습을 진행하였기 때문에 학습된 정책을 통해 모션이 생성되는 단일 강체 캐릭터에 정의된 범위 내의 외력을 주었을 시 본 논문에서 제안한 방법이 기존의 강화 학습[1]보다 더욱 견고하다는(robustness) 것을 실험적으로 보였다. 캐릭터에 작용된 외력 집합은 [100, 200, 300]이며 Figure 5는 외력이 작용했을 때 캐릭터의 상태를 보여주는 스냅 샷이다. 즉, 실험 결과를 통해 우리의 학습된 정책 네트워크는 정의된 외력 범위에서 균형을 잃지 않지만 [1]은 최소 200 N에서 쓰러지는 것을 확인할 수 있다. 추가적인 실험의 결과로 우리의 학습된 정책 네트워크에 의한 단일 강체 캐릭터는 최소 1500 N에서 균형을 잃었음을 확인할 수 있었다. 더하여 본 논문에서 제안한 방법의 견고함을 증명하기 위해 고정된 힘이 지속적으로 가해졌을 때 버틸 수 있는지에 대한 실험을 진행하였다. 이는 버틸 수 있는 힘이라도 지속적으로 가해졌을 때 안정적인 결과를 낼 수 있는지를 판단하기 위해 진행되었다. Table 1은 100 N의 외력이 정의된 시간만큼 가해졌을 때, 즉, 힘이 고정된 상태에서 시간을 변화시켜 운동량이 정해진 범위 내에서 가해졌을 때 각 캐릭터가 버틸 수 있는지에 대한 결과를 보여준다. 이를 통해 우리의 학습된 정책 네트워크는 0.4초 동안 100 N을 가했을 때 균형을 잘 유지하지만 [1]을 적용한 캐릭터는 쓰러진다는 것을 알 수 있다. 이를 통해 추가적으로 우리가 제안한 정책(policy) 네트워크가 더욱 견고하다는 것을 확인할 수 있다.



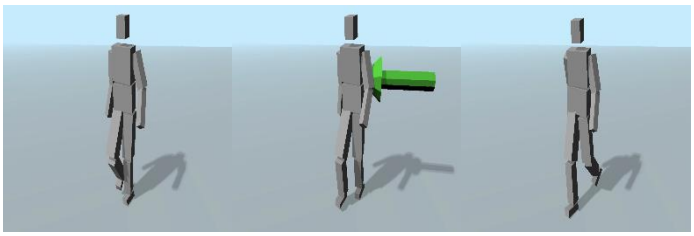
(a) Applied 100 N to SRB



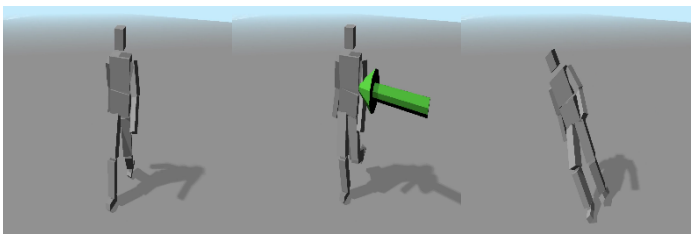
(b) Applied 200 N to SRB



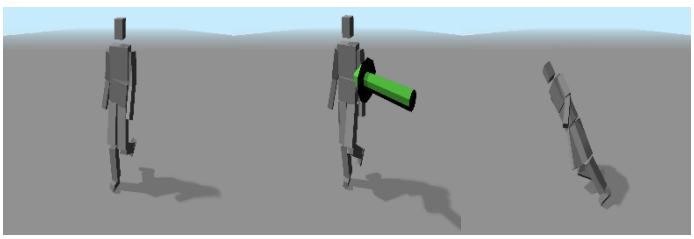
(c) Applied 300 N to SRB



(d) Applied 100 N to character trained by [1]



(e) Applied 200 N to character trained by [1]



(f) Applied 300 N to character trained by [1]

Figure 5: Snapshots of motions generated by our policy network and [1] when the external force is applied to character for 0.2 s. **Top-to-bottom:** 100 N, 200 N, 300 N by our policy network and 100 N, 200 N, 300 N by [1].

Table 1: Whether it is able to tolerate according to the different momentum for [1] and our policy network. The 'T' means that the character tolerates such momentum, and the 'F' means that the character fall down at such momentum.

Time(s)	Other [1]	Our policy network
0.01	T	T
0.1	T	T
0.2	T	T
0.3	T	T
0.4	F	T
1.0	F	T

10. 토론

본 논문에서는 단일 강체 캐릭터에 적용하는 제어 정책(control policy)을 학습할 수 있는 심층 강화 학습 프레임워크를 제시하였다. 학습된 정책에 현 스텝 t의 상태 벡터가 입력되면 액션 벡터를 얻게 된다. 이차 계획법 컨트롤러는 액션 벡터 성분들 중 단일 강체 캐릭터의 무게 중심의 이상 상대 속도를 나타내는 성분들과 레퍼런스 캐릭터의 무게 중심의 이상 속도 및 위치를 활용하여 단일 강체 캐릭터의 무게 중심에 적용될 접촉력과 토크를 계산한다. 계산된 힘들과 단일 강체 캐릭터의 접촉된 발의 위치를 통해 다음 단일 강체 캐릭터의 무게 중심의 좌표계가 계산된다. 계산된 단일 강체 캐릭터의 무게 중심의 좌표계는 다음 스텝 t+1에서 단일 강체 캐릭터의 발의 위치 계산에 활용된다. 제안된 방법처럼 단일 강체 캐릭터로 근사하여 학습을 진행하였으므로 우리는 [1]보다 학습 속도를 개선하였다. 견고성 또한 [1]을 통한 캐릭터보다 더 큰 외력과 운동량에서도 유지가 됨을 실험을 통해 알 수 있었다. 우리가 제안한 향상된 성능의 프레임워크는 후에 캐릭터의 모션을 레퍼런스 캐릭터의 모션에 따라 학습시킬 때 전신 동작이 아닌 중심 역학을 적용하여 강체로 근사하여 경로를 만들 때 사용될 수 있다. 즉, 중심 역학을 이용한 전신 동작의 모션 합성에서 우리가 제안한 방법은 경로 최적화를 대체하는 프레임워크로 적용될 수 있다. 우리가 제안한 프레임워크는 다음의 세가지 방향으로 발전시킬 수 있다.

첫번째는 현재 우리가 학습시킨 모션의 주체는 단일 강체 캐릭터로써 이는 완벽한 캐릭터의 형태라고 보기 어렵다. 이는 역으로 모델 예측 컨트롤러(model predictive control) 같은 방법을 활용하여 전신 동작으로의 모션 합성이 이루어질 필요성을 가지게 된다. 우리는 이후에 다양한 캐릭터에 대한 전신 동작으로의 모션 합성이 이루어질 수 있는 프레임워크로 확장할 계획이다. 두번째는 우리가 제안한 프레임워크에는 레퍼런스 캐릭터의 모션을 기준으로 접촉 타이밍을 미리 준비해야하는 과정이 필요하다는 것이다. 이는 견고함에 영향을 미친다. 접촉 타이밍이 정해진다는 것은 캐릭터가 외력을 받았을 때 좀 더 유연하게 적응하는 것에 방해가 되는 요인이기 때문이다. 외력의 반응에 더욱 견고하게 만들기 위해서 접촉 타이밍을 정해 놓지 않고 필요시에 발의 속도를 조절할 수 있는 방식으로 발전시킬 예정이다. 마지막으로 우리는 단일 강체 캐릭터의 무게 중심 높이를 레퍼런스 캐릭터의 무게 중심 높이와 비슷하게 고려하여 접촉 여부를 판단하였다. 이는 단일 강체의 무게 중심 높이를 랜덤화(randomizing) 하여 고정된 무게 중심 높이를 가진 단일 강체 캐릭터의 모션을 생성하는 것이 아닌 레퍼런스 캐릭터의 모션을 만족하면서 다양한 범위 내 무게 중심 높이를 가진 단일 강체 캐릭터의 모션을 생성하는 더욱 일반화(generalizing) 된 것으로 확장될 수 있다. 이는 앞서 언급한 다양한 종류의 캐릭터에 대한 전신 동작으로의 모션 합성에도 도움이 될 것이다.

References

[1] Peng, Xue Bin, et al. "Deepmimic: Example-guided deep reinforcement learning of physics-based character skills." *ACM Transactions on Graphics (TOG)*, vol. 37, no. 4, pp. 1–14, 2018.

[2] Coros, Stelian, Philippe Beaudoin, and Michiel Van de Panne. "Generalized biped walking control." *ACM Transactions on Graphics (TOG)*, vol. 29, no. 4, pp. 1–9, 2010.

[3] Ye, Yuting, and C. Karen Liu. "Optimal feedback control for character animation using an abstract model." *ACM SIGGRAPH 2010 papers*, pp. 1–9, 2010.

[4] Yin, KangKang, Kevin Loken, and Michiel Van de Panne. "Simbicon: Simple biped locomotion control." *ACM Transactions on Graphics (TOG)*, vol. 26, no. 3, pp. 105–es, 2007.

[5] Agrawal, Shailen, Shuo Shen, and Michiel van de Panne. "Diverse motion variations for physics-based character animation." *Proceedings of the 12th ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pp. 37–44, 2013.

[6] Ha, Sehoon, and C. Karen Liu. "Iterative training of dynamic skills inspired by human coaching techniques." *ACM Transactions on Graphics (TOG)*, vol. 34, no. 1, pp. 1–11, 2014.

[7] Wang, Jack M., et al. "Optimizing locomotion controllers using biologically-based actuators and objectives." *ACM Transactions on Graphics (TOG)*, vol. 31, no. 4, pp. 1–11, 2012.

[8] Da Silva, Marco, Yeuhi Abe, and Jovan Popović. "Simulation of human motion data using short-horizon model-predictive control." *Computer Graphics Forum*. Oxford, UK. Blackwell Publishing Ltd, vol. 27, no. 2, pp. 371–380, 2008.

[9] Lee, Yoonsang, Sungeun Kim, and Jehee Lee. "Data-driven biped control." *ACM SIGGRAPH 2010 papers*, pp. 1–8, 2010.

[10] Lee, Yoonsang, et al. "Locomotion control for many-muscle humanoids." *ACM Transactions on Graphics (TOG)*, vol. 33, no. 6, pp. 1–11, 2014.

[11] Mordatch, Igor, Emanuel Todorov, and Zoran Popović. "Discovery of complex behaviors through contact-invariant optimization." *ACM Transactions on Graphics (TOG)*, vol. 31, no. 4, pp. 1–8, 2012.

[12] Wampler, Kevin, Zoran Popović, and Jovan Popović. "Generalizing locomotion style to new animals with inverse optimal regression." *ACM Transactions on Graphics (TOG)*, vol. 33, no. 4, pp. 1–11, 2014.

[13] Hämäläinen, Perttu, Joose Rajamäki, and C. Karen Liu. "Online control of simulated humanoids using particle belief propagation." *ACM Transactions on Graphics (TOG)*, vol. 34, no. 4, pp. 1–13, 2015.

[14] Tassa, Yuval, Tom Erez, and Emanuel Todorov. "Synthesis and stabilization of complex behaviors through online trajectory optimization." *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, pp. 4906–4913, 2012.

[15] Lee, Yongjoon, et al. "Motion fields for interactive character locomotion." *ACM SIGGRAPH Asia 2010 papers*, pp. 1–8, 2010.

[16] Levine, Sergey, et al. "Continuous character control with low-dimensional embeddings." *ACM Transactions on Graphics (TOG)*, vol. 31, no. 4, pp. 1–10, 2012.

[17] Coros, Stelian, Philippe Beaudoin, and Michiel Van de Panne. "Robust task-based control policies for physics-based characters." *ACM SIGGRAPH Asia 2009 papers*, pp. 1–9, 2009.

[18] Peng, Xue Bin, Glen Berseth, and Michiel Van de Panne. "Dynamic terrain traversal skills using reinforcement learning." *ACM Transactions on Graphics (TOG)*, vol. 34, no. 4, pp. 1–11, 2015.

[19] Brockman, Greg, et al. "Openai gym." *arXiv preprint arXiv:1606.01540*, 2016.

[20] Duan, Yan, et al. "Benchmarking deep reinforcement learning for continuous control." *International conference on machine learning*. PMLR, pp. 1329–1338, 2016.

[21] Liu, Libin, and Jessica Hodgins. "Learning to schedule control fragments for physics-based characters using deep q-learning." *ACM Transactions on Graphics (TOG)*, vol. 36, no. 3, pp. 1–14, 2017.

[22] Peng, Xue Bin, Glen Berseth, and Michiel Van de Panne. "Terrain-adaptive locomotion skills using deep reinforcement learning." *ACM Transactions on Graphics (TOG)*, vol. 35, no. 4, pp. 1–12, 2016.

[23] Rajeswaran, Aravind, et al. "Learning complex dexterous manipulation with deep reinforcement learning and demonstrations." *arXiv preprint arXiv:1709.10087*, 2017.

[24] Teh, Yee Whye, et al. "Distral: Robust multitask reinforcement learning." *arXiv preprint arXiv:1707.04175*, 2017.

[25] Orin, David E., Ambarish Goswami, and Sung-Hee Lee. "Centroidal dynamics of a humanoid robot." *Autonomous robots*, vol. 35, no. 2, pp. 161–176, 2013.

[26] Dai, Hongkai, Andrés Valenzuela, and Russ Tedrake. "Whole-body motion planning with centroidal dynamics and full kinematics." *2014 IEEE-RAS International Conference on Humanoid Robots*. IEEE, pp. 295–302, 2014.

[27] Winkler, Alexander W., et al. "Gait and trajectory optimization for legged systems through phase-based end-effector parameterization." *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 1560–1567, 2018.

[28] Kwon, Taesoo, Yoonsang Lee, and Michiel Van De Panne. "Fast and flexible multilegged locomotion using learned centroidal dynamics." *ACM Transactions on Graphics (TOG)*, vol. 39, no. 4, pp. 1–46, 2020.

[29] Xie, Zhaoming, et al. "GLiDE: Generalizable Quadrupedal Locomotion in Diverse Environments with a Centroidal Model." *arXiv preprint arXiv:2104.09771*, 2021.

[30] Schulman, John, et al. "Proximal policy optimization algorithms." *arXiv preprint arXiv:1707.06347*, 2017.

[31] Abe, Yeuhi, Marco Da Silva, and Jovan Popović. "Multiobjective control with frictional contacts." *Proceedings of the 2007 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pp. 249–258, 2007.

[32] da Silva, Marco, Yeuhi Abe, and Jovan Popović. "Interactive simulation of stylized human locomotion." *ACM SIGGRAPH 2008 papers*, pp. 1–10, 2008.

[33] Kwon, Taesoo, and Jessica K. Hodgins. "Momentum-mapped inverted pendulum models for controlling dynamic human motions." *ACM Transactions on Graphics (TOG)*, vol. 36, no. 1, pp. 1–14, 2017.

[34] Ellis, Jane, et al. "CDM: Taking stock and looking forward." *Energy policy*, vol. 35, no. 1, pp. 15–28, 2007.

〈저자소개〉

안 제 원

- 2015-2019 한양대학교 기계공학부 학사
- 2020-현재 한양대학교 지능융합학과 석사
- 관심 분야 : Physically-Based Character Control, Deep Reinforcement Learning
- https://orcid.org/0000-0002-4151-5372

구 태 홍

- 2006-2015 한양대학교 컴퓨터공학부 학사
- 2015-2018 한양대학교 지능형 로봇학과 석사
- 2018-현재 한양대학교 컴퓨터소프트웨어학부 박사
- 관심분야 : Physically-Based Character Control, Optimal Control Theory, Artificial Intelligence
- https://orcid.org/0000-0003-0590-153X

권 태 수

- 1996-2000 서울대학교 전기컴퓨터공학부 학사
- 2000-2002 서울대학교 전기컴퓨터공학부 석사
- 2002-2007 한국과학기술원 전산학전공 박사
- 관심 분야: Physics-based models, Machine learning techniques.
- https://orcid.org/0000-0002-9253-2156