

참조 동작을 사용하지 않는 근골격계 캐릭터의 보행 모션 학습*

김민관⁰, 이윤상
한양대학교 컴퓨터소프트웨어학과
{palkan21,yoonsanglee}@hanyang.ac.kr

Learning Locomotion of a Musculoskeletal Character without Reference Motions

Minkwan Kim⁰, Yoonsang Lee
Department of Computer Software, Hanyang University

요약

캐릭터가 참조 동작을 따라하도록 각 관절에 필요한 적절한 토크의 크기를 강화학습으로 찾는 다양한 연구들이 진행되고 있다. 사람의 관절은 직접 힘을 낼 수 없고 근육에 의해 관절이 움직이기 때문에 근육 시뮬레이션이 토크 시뮬레이션보다 더 자연스러운 동작을 생성할 수 있다. 이 연구에서는 참조 동작 없이 근육 시뮬레이션으로 사람 캐릭터의 보행 모션을 학습한다.

1. 서론

캐릭터 애니메이션 분야에서 물리 시뮬레이션과 강화학습을 사용해서 참조 동작을 따라하는 정책을 학습하는 연구들이 최근 다양하게 진행되었다. 물리 시뮬레이션은 각 관절에 토크를 적용하는 방식으로 캐릭터를 움직이는데, 실제 사람은 관절에서 직접 토크를 만들어 낼 수 없다. 실제 사람처럼 근육의 수축과 이완을 이용해서 관절의 움직임을 만들어내는 근육 시뮬레이션을 통해서 더 자연스러운 동작을 만들어낸 연구[1], [2]들이 있다. 이 연구에서는 근육 시뮬레이션을 사용해서 캐릭터의 보행 모션을 학습하는 모델을 제안한다. 참조 동작을 따라하는 대신 근육의 소모 에너지를 최소화하는 목표를 포함하는 강화학습을 통해서 자연스러운 보행 동작을 만들고 그 때 각 근육의 활성화 정도를 분석할 수 있다. 또한 근육의 활성화 정도를 강화학습으로 바로 출력해서 사용하는 방식과, 토크 시뮬레이션에 필요한 목표 자세를 출력한 뒤 최적화 문제를 풀어서 근육의 활성화 정도를 계산해서 시뮬레이션 하는 방식을 비교한다.

2. 근육 시뮬레이션

근육 시뮬레이션에서 사용하는 근육은 근섬유와 힘줄로 구성된다. 근섬유는 능동적인 부분과 수동적인 부분으로 나누어진다. 근육은 캐릭터의 몸체들 사이에서 경로 점(path point)로 연결되어 있고, 각 경로 점 사이에서 서로 당기는 힘이 작용해서 관절을 움직인다. 근육이 낼 수 있는 힘은 근섬유의 길이, 속도, 근육의 수축 및 이완 정도에 의해 결정된다. 근육의 수축 정도는 0(이완)에서 1(최대 수축)사이의 활성화 값으로 표현된다. 근육의 힘을 계산하는 수식은 아래와 같다[3].

$$f_{mt} = \cos \alpha \cdot (act \cdot g_{al}(l_m) \cdot g_v(\dot{l}_m) + g_{pl}(l_m)) = g_t(l_t)$$

l_m , \dot{l}_m 은 근섬유의 길이와 속도, α 는 근섬유와 힘줄 사이의 각도, act 는 활성화 값을 의미한다. g_{al} , g_v 는 근섬유의 능동적인 부분이 내는 힘, g_{pl} 은 수동적인 부분이 내는 힘, g_t 는 힘줄이 내는 힘을 계산하는 함수이다. 근육 시뮬레이션을 위해서 우선 근육의 길이, 속도, 활성화 정도 값을 이용해서 각 근육이 낼 수 있는 힘을 계산한다. 그리고 그 힘을 경로 점 사이에서 서로 당기는 방향으로 적용한다. 캐릭터에 힘이 적용되면 캐릭터의 각 관절이 움직이고 캐릭터의 상태가 변한다. 바뀐 캐릭터의 상태에 맞게 근육의 총 길이가 변하고, 경로 점의 위치가 수정된다. 바뀐 근육의 길이를 이용해서 근섬유의 속도를 계산하고, 근섬유의 속도를 시간에 대해 적분해서 근섬유의 길이를 계산한다. 이 때 근육의 속도는 근섬유의 힘과 힘줄의 힘이 같아지도록 계산된다.

3. 강화 학습

가만히 서있는 캐릭터가 넘어지지 않고 앞으로 나아가는 동작을 참조 동작의 도움 없이 바로 학습하는 것이 어렵기 때문에, 해당 연구에서는 커리큘럼 러닝[4]방식으로 학습을 진행했다. 커리큘럼 러닝은 쉬운 목표에서 먼저 학습한 뒤, 목표를 달성하면 점진적으로 어려운 목표로 진행하는 방식이다. 캐릭터가 앞으로 나아가는 데 도움을 주기 위해서 뒤에서 앞으로 밀어주는 힘을 적용했고, 캐릭터가 좌우로 넘어지는 것을 막기위해서 캐릭

* 구두 발표논문, 요약논문(Extended Abstract)

* 본 연구는 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원을 받아 수행되었음 (NRF-2019R1C1C1006778).

터의 좌우 방향으로 힘을 적용했다. 힘은 캐릭터의 무게 중심에 적용되었으며, stable PD control (SPD) [5]을 이용해서 계산되었다. 학습 초기에는 비례계수(P gain)과 미분계수(D gain)을 각각 2000으로 설정했고, 현재 단계에서 6초이상 넘어지지 않고 버틸 경우 두 계수를 0.75배 감소시키며 다음 단계로 넘어갔다.

강화학습 정책은 환경에서 상태정보를 받아서 다음 행동을 출력하고, 보상 함수를 높이는 방향으로 가중치 값을 수정한다. 상태정보는 캐릭터의 각 관절의 각도와 각속도를 포함하며, 각도와 각속도는 캐릭터의 루트(골반)의 프레임에 대하여 표현된다.

$$obs = \{q, \dot{q}\} \in \mathbb{R}^{62}$$

강화 학습의 보상함수는 총 6가지 값으로 구성된다

$$reward = r_{up} + r_{ori} + r_{dev} + r_{vel} + r_{eng} + r_{alive}$$

r_{up} , r_{ori} , r_{dev} 는 캐릭터의 상체 밸런스를 유지하기 위한 값들이다. r_{vel} 는 캐릭터가 목표 속도로 나아가도록 하는 값이고, r_{eng} 는 캐릭터의 에너지 소모를 최소화하는 값이다. r_{alive} 는 캐릭터가 넘어지지 않고 최대한 오래 버티도록 하는 상수이다. 각 보상의 가중치는 실험을 통해서 보상의 값들이 비슷하게 계산되도록 조절했다.

해당 연구에서는 강화학습의 출력으로 근육 활성화 값을 사용하는 것과 각 관절의 목표 자세를 사용하는 것 두 가지 방식을 비교했다. 전자의 경우 정책이 출력하는 행동은 캐릭터가 가진 120개 근육의 활성화 정도 값이며, 근육 활성화 값으로 바로 근육 시뮬레이션을 진행했다. 후자의 경우 정책이 출력하는 행동은 캐릭터의 골반의 6자유도를 제외한 25차원의 목표 자세 값이다. 캐릭터의 현재 자세를 목표 자세로 바꾸기 위한 목표 토크를 계산한 뒤, 목표 토크를 발생시키는 근육의 각 활성화 정도 값을 최적화 문제를 풀어서 계산해서 근육 시뮬레이션을 진행했다. 최적화 문제 수식은 아래와 같다.

$$objective\ function = \|\tau_{desired} - \tau_{activation}\|_2^2$$

$$0 \leq activation \leq 1$$

$$\tau_{activation} = J^T V_a C F_{mt}$$

0에서 1사이의 activation(근육 활성화 값) 중에서 목표 토크와 차이가 가장 적은 토크를 만드는 activation을 찾는다. Activation으로 계산된 근육의 힘은 120차원이고, 캐릭터의 관절에 적용되는 토크는 31차원이기 때문에 $J^T V_a C$ 를 곱해서 글로벌 좌표계에서 표현된 근육의 힘을 각 관절 좌표계로 변경했다. $V_a C$ 는 경로 점 사이의 방향 벡터에 대한 행렬이고, J^T 는 글로벌 좌표계의 힘을 관절 좌표계의 힘으로 변경하는 행렬이다.

4. 실험 결과

해당 연구에서는 총 4가지 학습 방식의 결과를 비교했다. 첫번째로 강화학습 정책으로 근육 활성화 값을 출력해서 근육 시뮬레이션으로 학습했다. 두번째로 강화학습 정책으로 출력한 목표 자세를 최적화 문제로 근육 활성화 값으로 바꿔서 근육 시뮬레이션으로 학습했다. 세번째로 목표 자세에 대해 SPD로 계산한 토크로 시뮬레이

션 하는 것과 최적화 문제를 풀어서 근육 시뮬레이션 하는 것을 번갈아 가면서 학습했다. 네번째로 [2]를 참고해서 최적화 문제를 푸는 네트워크를 지도학습으로 학습한 뒤, 네트워크가 출력한 근육 활성화 값으로 근육 시뮬레이션으로 학습했다. 이 때 지도 학습의 손실 함수로 최적화 문제의 objective function을 사용했다.

Figure 1은 순서대로 네 가지 학습 방식의 결과이다. 첫번째는 넘어지지 않고 앞으로 나가는 결과가 나왔으나 걷는 모습이 실제 사람의 동작과는 다르다. 두번째는 팔을 들고 넘어지는 결과가 나왔다. 세번째는 동작은 자연스럽지만 첫 걸음 이후로 균형을 잃고 금방 넘어졌다. 네번째는 제자리에서 넘어졌다.

모든 결과에는 학습이 수렴된 시점의 커리큘럼에 따른 외력이 루트(골반)에 적용되었다. 최적화 문제를 사용하는 방식은 학습 도중 정책이 출력한 목표 자세가 실제 사람의 자세와는 거리가 먼 경우가 많았다. 이 경우 최적화 문제의 오차가 커지면서 학습이 잘 진행되지 않는 결과가 나온 것으로 보인다. 네 번째 방식은 학습 시간이 충분하지 않았을 수 있다. 120개 근육의 근섬유 길이 상태 정보에 포함시켜 학습을 해보기도 했는데, 늘어난 입력 차원수에 따라 네트워크 크기와 학습 시간이 늘어난 것에 비해 결과의 향상은 미미하였다.

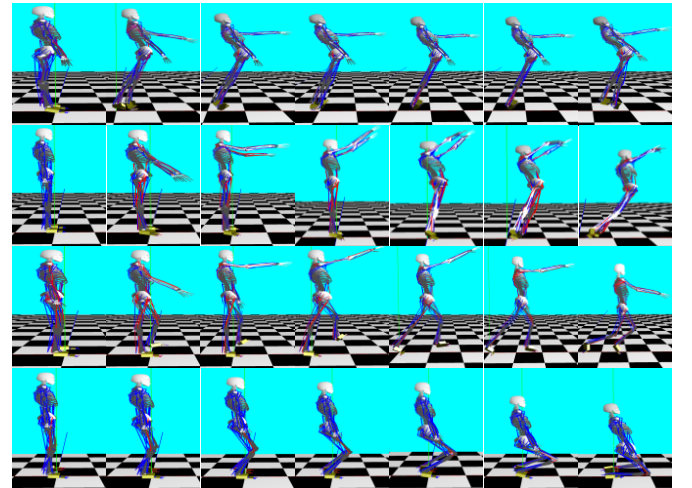


Figure 1: Result of 4 learning methods, top to bottom.

참고문헌

- [1] Y.S.Lee, M.S.Park, T.S.Kwon, J.H.Lee, Locomotion control for many-muscle humanoid, ACM Transactions on Graphics, 33(6), 2014.
- [2] S.H.Lee, M.S.Park, K.M.Lee, J.H.Lee, Scalable muscle-actuated human simulation and control, ACM Transactions on Graphics, 38(4), 2019.
- [3] Thelen, D.G, Adjustment of muscle mechanics model parameters to simulate dynamic contractions in older adults. ASME Journal of Biomechanical Engineering, 125(1), 2003.
- [4] W.Yu, G.Turk, K.Liu, Learning Symmetric and Low-energy Locomotion, ACM Transactions on Graphics 37(4), 2018.
- [5] J.Tan, K.Liu, G.Turk, Stable Proportional-Derivative Controllers, IEEE Computer Graphics and Applications, 31, 2011.