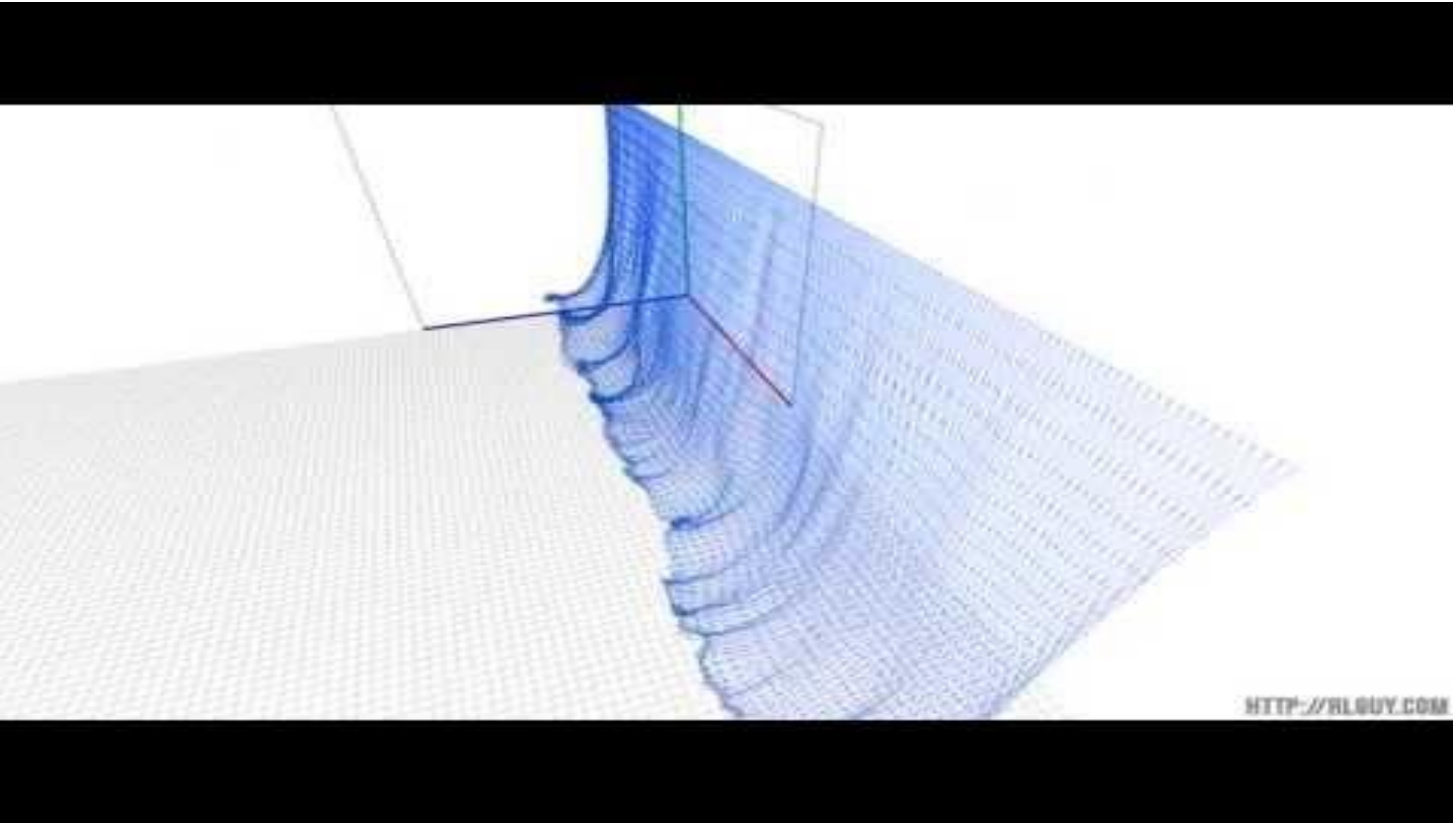
COMPUTER SCIENCE Capstone PBL
- Character Motion Synthesis and Character Control

5 - Particle Dynamics,
Physics-Based Character Control

Yoonsang Lee

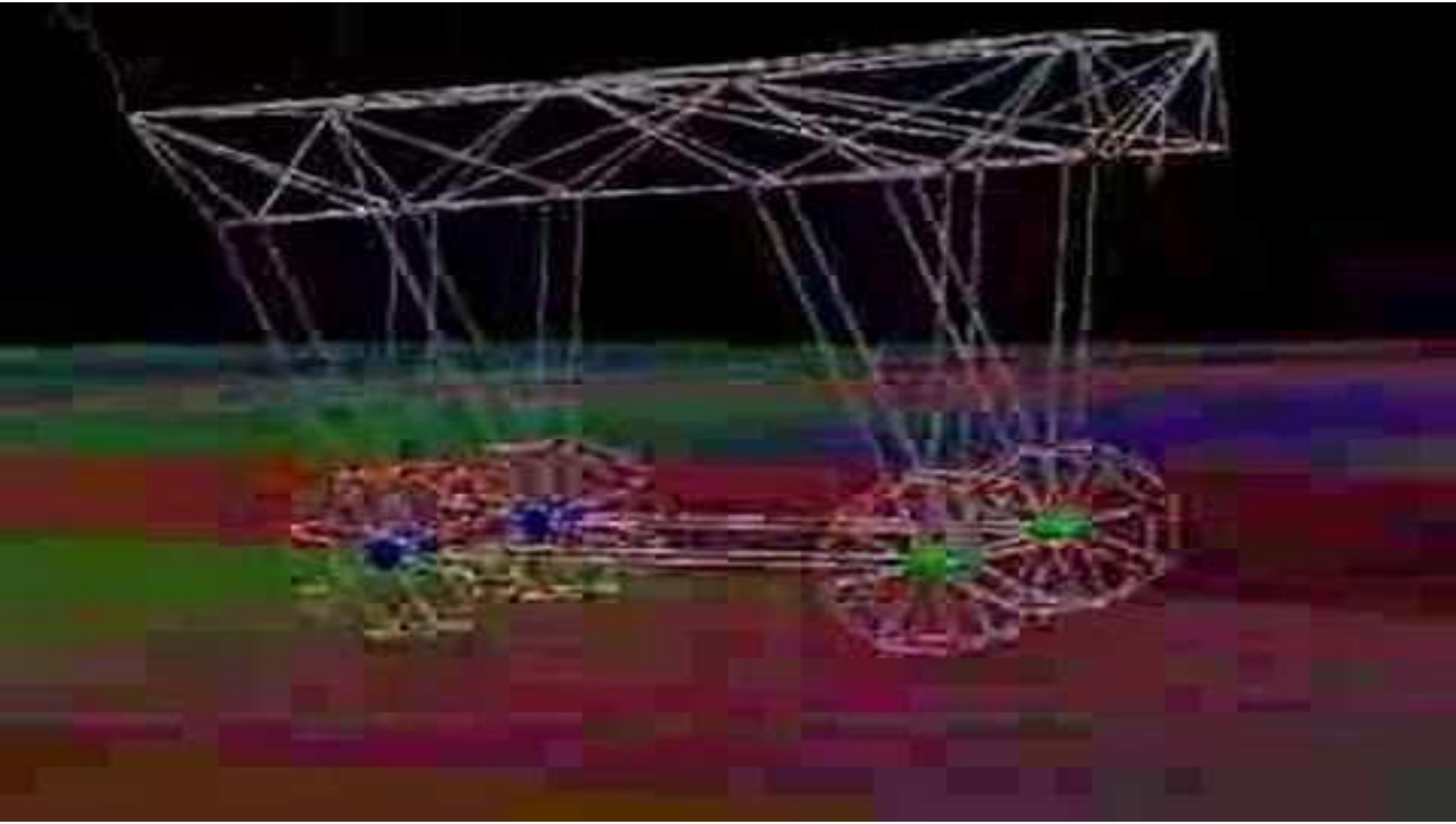
Fall 2022

Example 1



https://youtu.be/TuQwzNKcUtk

Example 2



https://youtu.be/qmLBCPSSCNQ

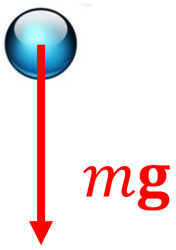
Today's Topics

- Physics Basics
- Differential Equation Basics
- Particle Dynamics
- Intro to Implicit Methods
- Physics Simulation Steps
- Physics-Based Character Control Problem
- Intro to DRL-Based Physics-Based Character Control

Physics Basics

Physics Basics

- Mass (질량)
  - 물체가 가지고 있는 고유한 양으로, '물체가 움직이기 어려운 정도'를 나타낸다. kg(킬로그램) 등의 단위를 사용한다.
  - 오른쪽 그림에서 스칼라 m으로 표현.

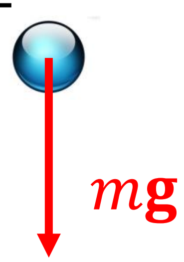


- Gravitional Acceleration (중력가속도)
  - 중력에 의해 발생하는 가속도이다.
  - 지구의 중력가속도는 약 9.8 m/s^2 으로, 공기 저항이 전혀 없다면 자유 낙하하는 물체의 속력이 1초마다 약 9.8 m/s 씩 증가한다는 의미이다.
  - 오른쪽 그림에서 벡터 **g**로 표현.

Physics Basics

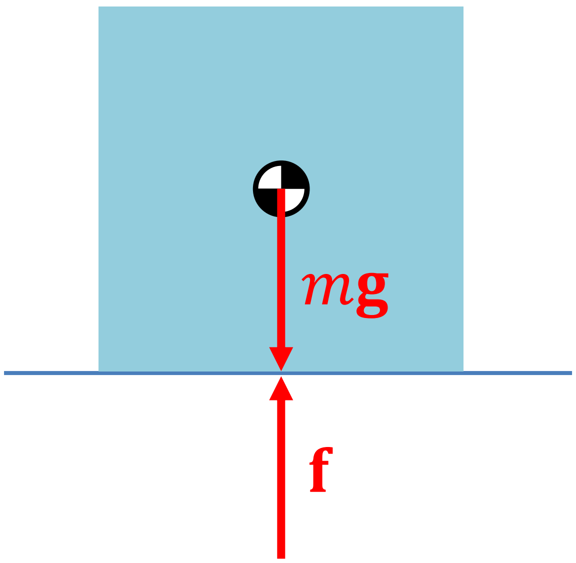
- Force (힘)
  - 물체의 속도를 변화시키는 요인이 되는 물리량.
  - **F** = m**a**
  - 보통 N (뉴턴)을 단위로 사용한다. 1 N = 1 kg * 1 m/s²

- Weight (무게)
  - 질량과 중력가속도의 곱(m**g**)으로, 물체에 가해지는 중력의 크기를 의미한다.
  - 질량은 물체의 고유한 양이나, 무게는 중력에 따라 달라진다.



Physics Basics

- Ground Reaction Force (GRF) (지면 반력)
  - 물체를 지구의 중심으로 끌어당기는 중력에 반해서, 물체가 지면 위에 버티고 있을 수 있도록 지면과의 접촉점에 작용하는 접촉력 (contact force).
  - 지구로부터 물체에 가해지는 힘으로 생각하면 물리적으로 이해하기 편하다. 만일 지면 위에 가만히 놓여 있는 물체가 있다면, 그 [물체의 무게]와 [물체에 작용하는 지면 반력]은 크기는 같지만 방향은 서로 반대인 힘 벡터이다.
  - 사람이 땅으로 뚫고 들어가지 않고 땅 위에 서 있을 수 있는 것은 지면 반력이 작용하기 때문이라고 말할 수 있으며, 한편으로는 땅으로 뚫고 들어가지 못하기 때문에 지면 반력이 생길 수 밖에 없다고도 말할 수도 있다.
  - 오른쪽 그림에서 벡터 f로 표현.



Physics Basics

- Linear Momentum (선운동량)
  - 질량이 m이고, **v**의 속도로 움직이는 물체는 **P**=m**v** 만큼의 선운동량을 갖는다.
  - **P**(t) = m**v**(t), $\dot{\mathbf{P}}(t) = m\dot{\mathbf{v}}(t) = \mathbf{F}(t)$

- Law of Momentum Conservation (운동량 보존의 법칙)
  - 어떤 계(system)의 외부에서 힘(force)이 가해지지 않는다면 (closed system), 계의 총 운동량은 변하지 않는다.

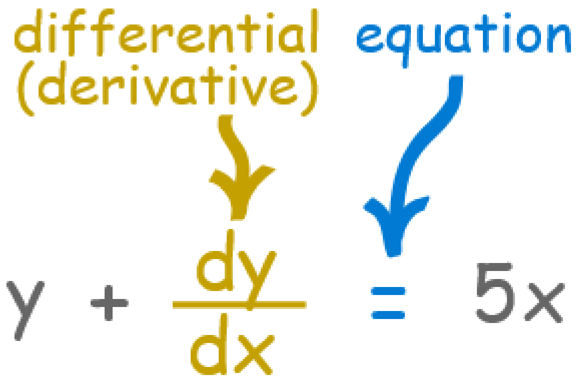
- Impulse (충격량)
  - 일정 시간 동안의 운동량의 변화. 운동량과 단위가 같다.
  - 어떤 물체가 시각 t_0에 운동량 $\mathbf{p_0}$, 시각 t_1에 운동량 $\mathbf{p_1}$을 가지는 경우 물체가 t_0부터 t_1까지 받은 충격량 **I**:

$$\mathbf{I} = \mathbf{p}_1 - \mathbf{p}_0 = \int_{t_0}^{t_1} \dot{\mathbf{p}}(t)\, dt \qquad \mathbf{I} = \int_{t_0}^{t_1} \mathbf{F}(t)\, dt$$

Differential Equation Basics

Differential Equation

- : An equation that relates one or more (unknown) functions and their derivatives. [Wikipedia]



- Solving a differential equation
- → Finding the unknown function y(x)

- Good intro reference:
  – https://www.mathsisfun.com/calculus/differential-equations.html

Why Differential Equation?

- Differential equations describes **how things change.**

- Play a prominent role in many disciplines such as engineering, physics, economics, biology, etc.

- Example: Continuous compound interest (연속복리)
  - **t** for time, **r** for the interest rate and **V** for the current money
  - $\frac{dV}{dt} = rV$
  - V(t) = ?

Examples

- Example: Population
  - the population **N** at any time **t**
  - the growth rate **r**
  - the maximum population that the food can support **k**
  - $\frac{dN}{dt} = rN(1 - \frac{N}{k})$ (Verhulst equation)
  - N(t) = ?

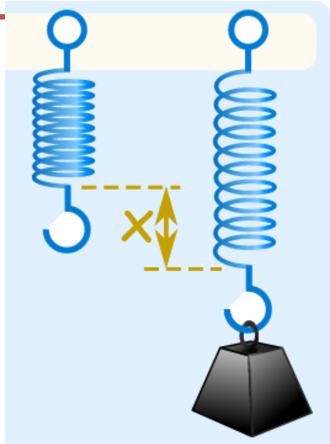
- Example: Equation of motion
  - $F = ma \rightarrow F = m\frac{d^2x}{dt^2}$
  - x(t) = ?

Solving Differential Equation

- Only the simplest differential equations are solvable by explicit formulas by many different kind of "tricks".

- Example: Continuous compound interest
  - **t** for time, **r** for the interest rate and **V** for the current money
  - $\frac{dV}{dt} = rV$
  - $V(t) = V(0)e^{rt}$ (by Separation of Variables)

Solving Differential Equation



- Example: A spring
  - $F = -kx \rightarrow m\frac{d^2x}{dt^2} = -kx$
  - ...
  - $x(t) = Asin(wt)$
  - https://www.mathsisfun.com/physics/ispring.html

- Can we get analytic solutions for ...?
  - Connecting multiple nodes with multiple springs?
  - Adding ground contact force / friction force?
  - Adding viscous drag force?
  - Interacting with objects / characters?
  - ...

Solving Differential Equation

- For most differential equations, analytic solutions are not available.

- Solutions may be approximated numerically using computers.

- That's what we're about to do.

"Physically Based Modeling" Course Notes

- The remainder of today's lecture is based on excellent SIGGRAPH course notes “Physically Based Modeling: Principles and Practice,” by David Baraff and Andrew Witkin.

- https://www.cs.cmu.edu/~baraff/sigcourse/

- " Slides"만으로 이해가 잘 안 되는 부분은 함께 제공되는 "Lecture Note"를 읽고 이해할 수 있도록 한다.

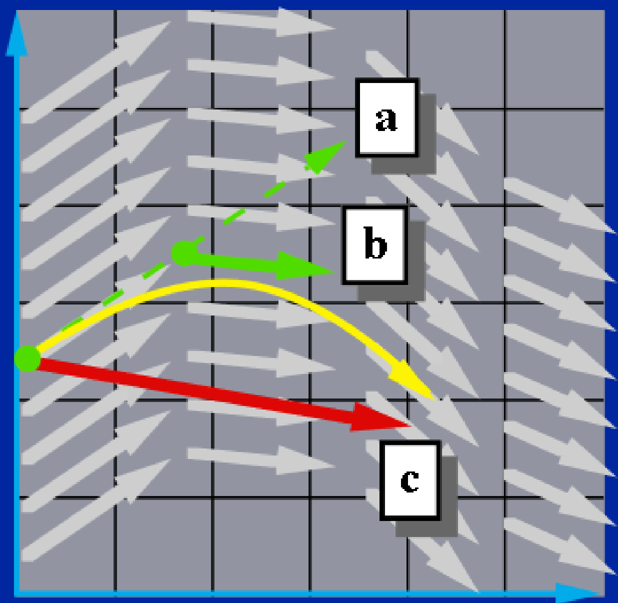
Differential Equation Basics

- Slides
  - https://www.cs.cmu.edu/~baraff/sigcourse/slidesb.pdf

- Lecture Notes
  - https://www.cs.cmu.edu/~baraff/sigcourse/notesb.pdf

Midpoint Method - Corrections for Page 9

The Midpoint Method



a. Compute an Euler step

$$\Delta x = \Delta t \, f(x,t)$$

b. Evaluate f at the midpoint

$$f_{mid} = f\left(\underline{x + \frac{\Delta x}{2},\ t + \frac{\Delta t}{2}}\right)$$

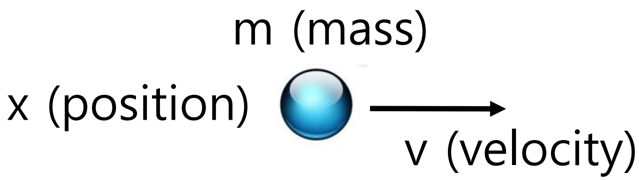
c. Take a step using the midpoint value

$$x(t + \Delta t) = x(t) + \Delta t \, f_{mid}$$

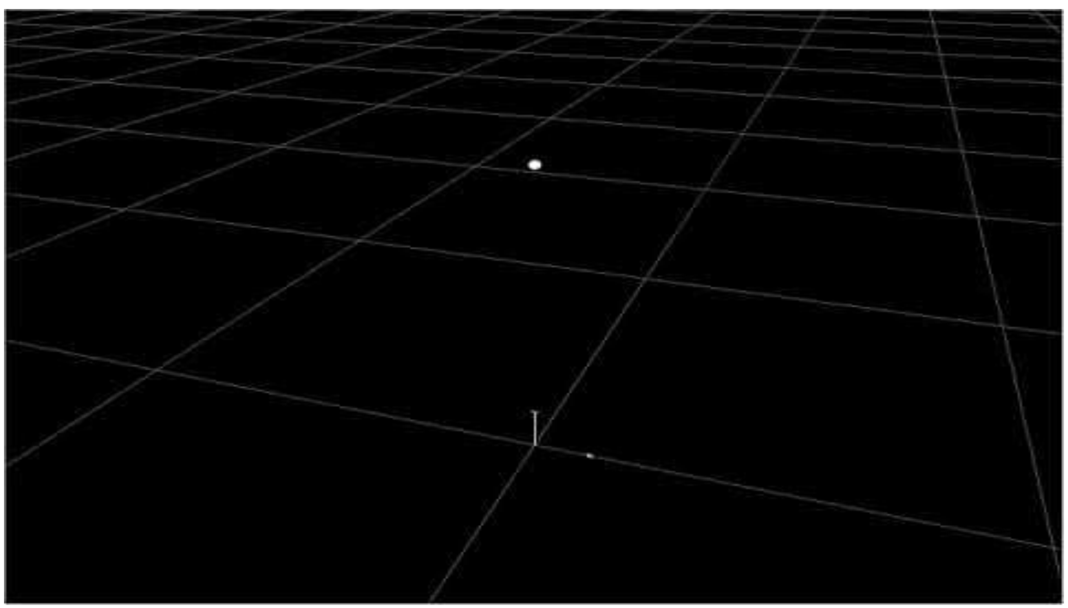
Particle Dynamics

Particle

- Particles are objects that have mass, position, and velocity, and respond to forces, but that have no spatial extent and rotational movement.



$$\mathbf{F} = m\mathbf{a}$$



(applied force : $\mathbf{F} = m\mathbf{g}$)

t:0.00	x:1.00	v:-0.33
t:0.03	x:0.99	v:-0.65
t:0.07	x:0.97	v:-0.98
t:0.10	x:0.93	v:-1.31
t:0.13	x:0.89	v:-1.63
t:0.17	x:0.84	v:-1.96
t:0.20	x:0.77	v:-2.29
t:0.23	x:0.70	v:-2.61
t:0.27	x:0.61	v:-2.94
t:0.30	x:0.51	v:-3.27

…

Particle Dynamics

- Slides
  - https://www.cs.cmu.edu/~baraff/sigcourse/slidesc.pdf
- Lecture Notes
  - https://www.cs.cmu.edu/~baraff/sigcourse/notesc.pdf

Damped Spring - Reference for Page 16

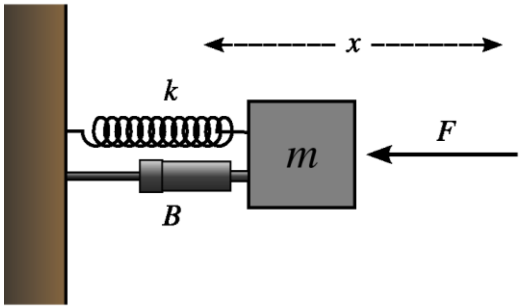
Force Law:

$$\mathbf{f}_1 = -\left[k_s(|\Delta\mathbf{x}| - r) + k_d\left(\frac{\Delta\mathbf{v}\cdot\Delta\mathbf{x}}{|\Delta\mathbf{x}|}\right)\right]\frac{\Delta\mathbf{x}}{|\Delta\mathbf{x}|}$$

$$\mathbf{f}_2 = -\mathbf{f}_1$$

- Δx: p1.x - p2.x
- Δv: p1.v - p2.v
- r: rest (default) length of a spring

- A spring in 1D space with zero rest length: $F_s = -kx + -c\dot{x}$



Discussion #1

- Go to https://www.slido.com/
- Join #**pbl-ys**
- Click "Polls"

- 아래의 형식으로 적어서 제출할 것.
  - **이름**: 자신의 의견 blah blah ...

Particle Dynamics - Review

- Slides
  - https://www.cs.cmu.edu/~baraff/sigcourse/slidesd.pdf
    page 1 -- page 10

- Lecture Notes
  - https://www.cs.cmu.edu/~baraff/sigcourse/notesd1.pdf
    Part 1, Section 1

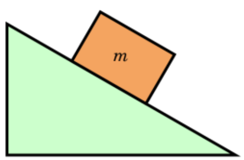
Friction Force

- If a collision is detected for a particle or a particle is in contact, add friction force to the particle.

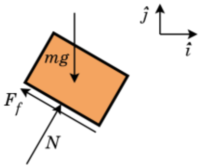
- Coulomb friction model: The frictional force is proportional to the applied normal force.

- $$\mathbf{F}_{\mathrm{friction}} = -\mu|\mathbf{F}_{\mathrm{n}}|\frac{\mathbf{V}_{\mathrm{t}}}{|\mathbf{V}_{\mathrm{t}}|}$$
  - μ: friction coefficient
    - https://en.wikipedia.org/wiki/Friction#Approximate_coefficients_of_friction
  - $\mathbf{F}_n$: normal force, $\mathbf{V}_t$: tangential velocity of the particle

A block on a ramp



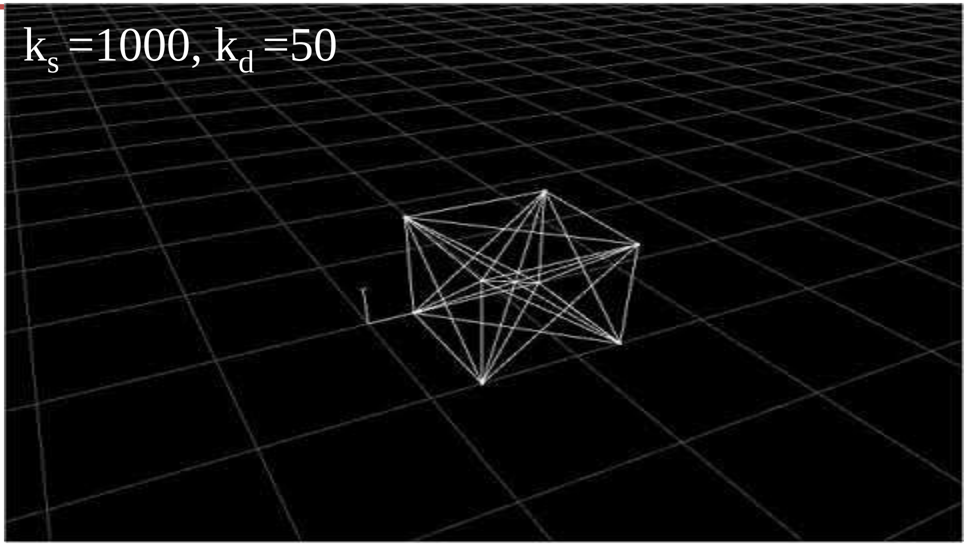
Free body diagram of just the block



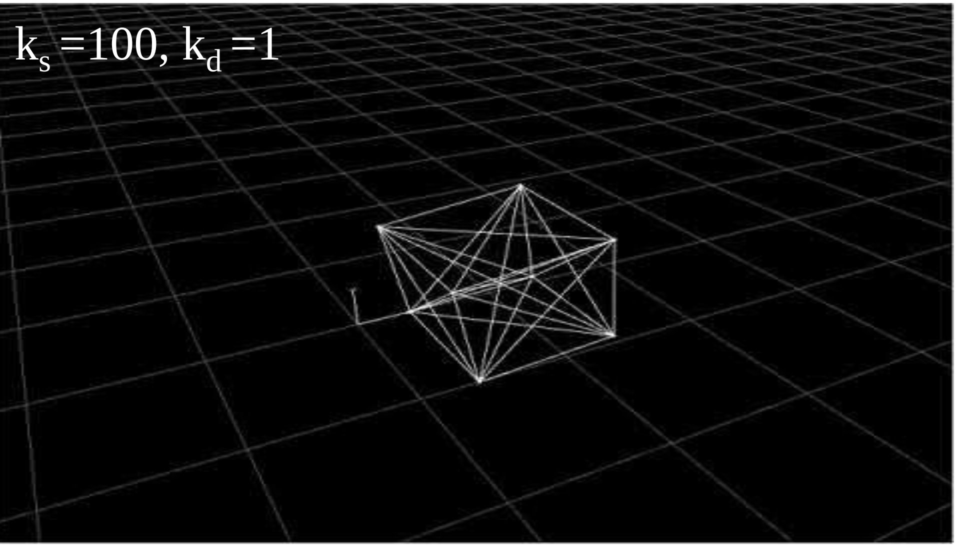
- You need to set the velocity a particle to zero if the speed of the particle is less than some threshold. Otherwise, the particle would not make a complete stop.

Mass-Spring Model Demo

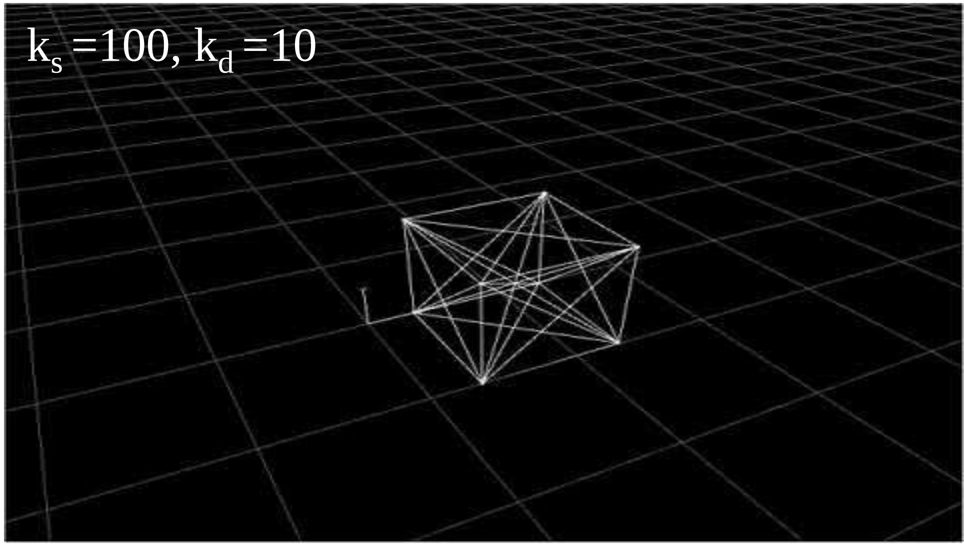
- Simulation with gravity force, spring force, and ground contact force (with friction force)



https://youtu.be/cdHc_eOEpFM

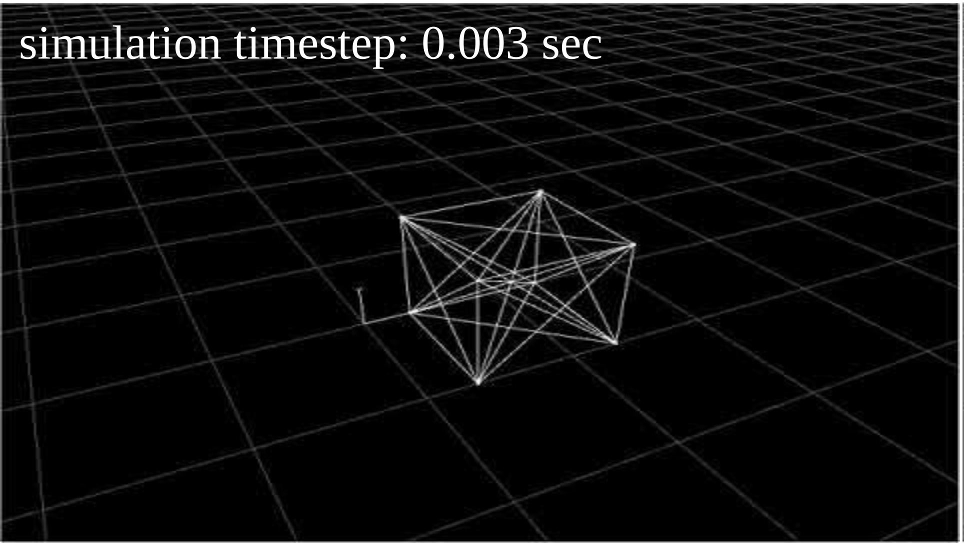


https://youtu.be/MnKK4p8YZ10

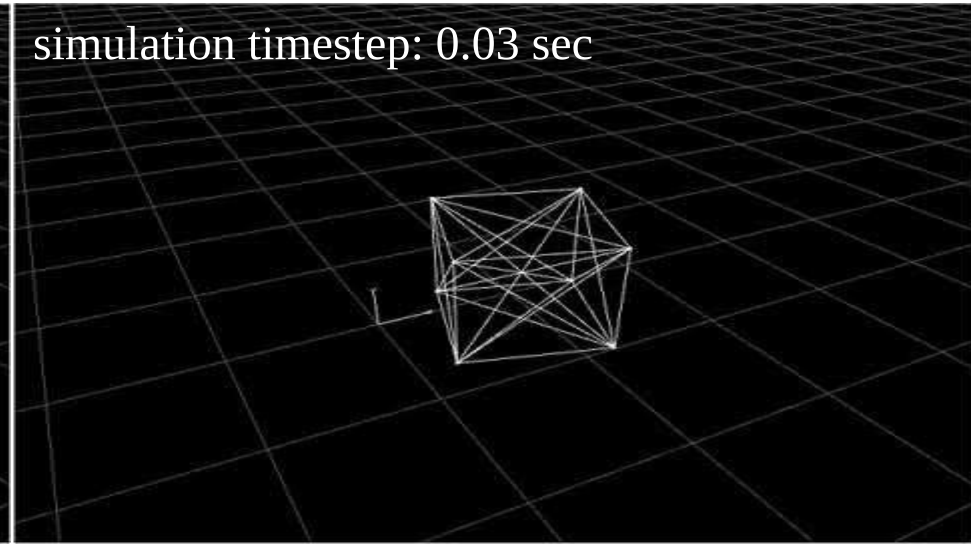


https://youtu.be/U0bE3fiuhUY

Stability & Time-step Size

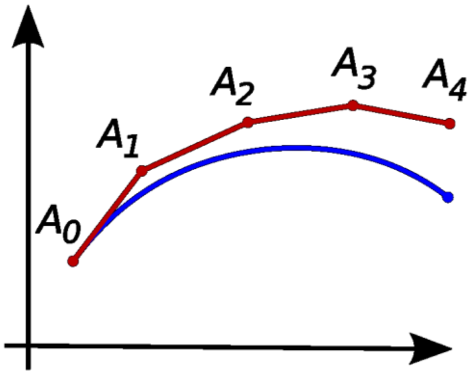


https://youtu.be/cdHc_eOEpFM

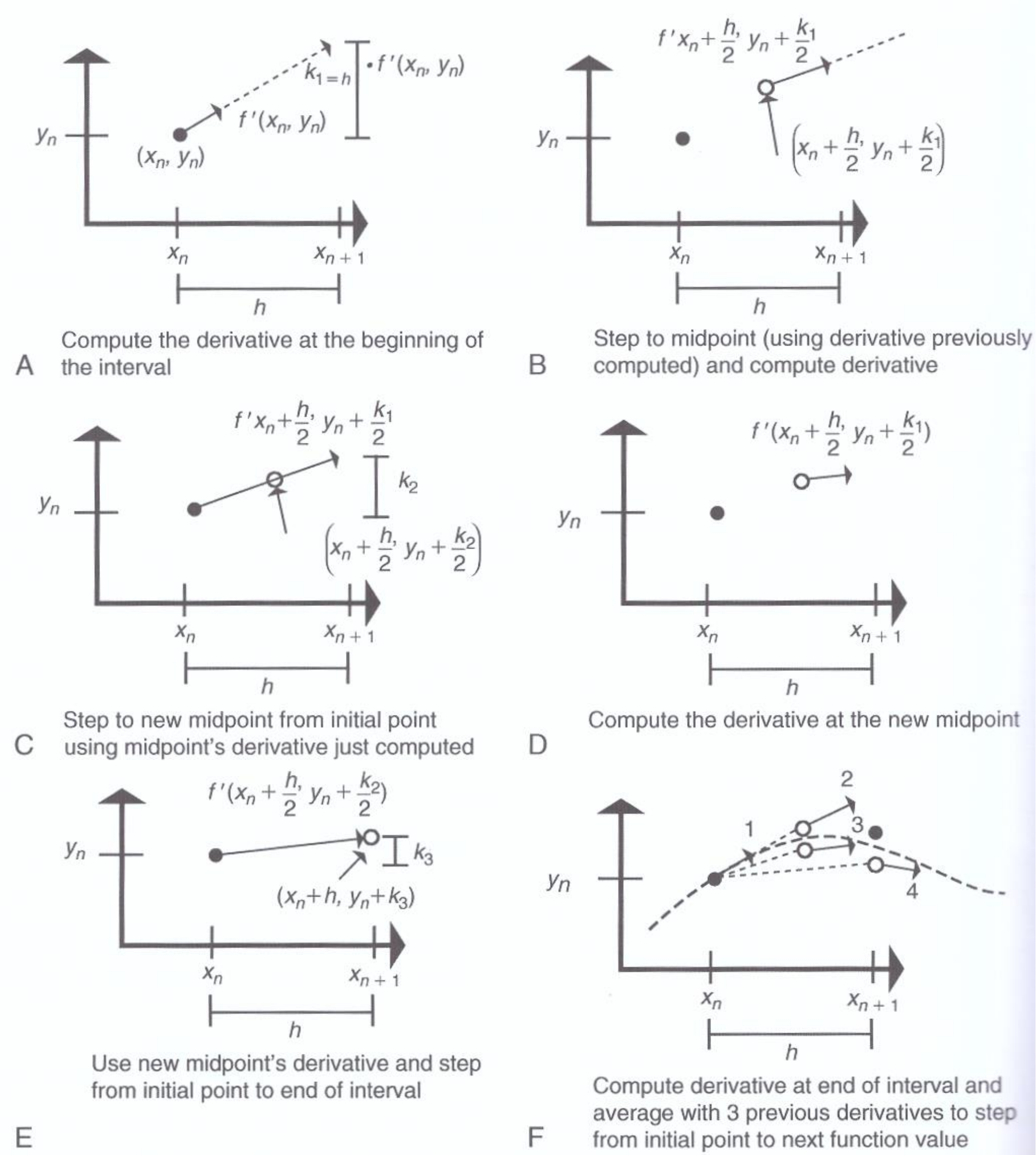


https://youtu.be/98A1Il_NVhc

- Too big step size → simulation diverges
- Too small step size → simulation goes very slow
- Depends on system stiffness & integration methods



Popular Integration Method 1 - Runge-Kutta 4th Order Method



$$k_1 = hf'(x_n, y_n)$$

$$k_2 = hf'\left(x_n + \frac{h}{2}, y_n + \frac{k_1}{2}\right)$$

$$k_3 = hf'\left(x_n + \frac{h}{2}, y_n + \frac{k_2}{2}\right)$$

$$k_4 = hf'(x_n + h, y_n + k_3)$$

$$y_{n+1} = y_n + \frac{k_1}{6} + \frac{k_2}{3} + \frac{k_3}{3} + \frac{k_4}{6}$$

Intro to Implicit Methods

Implicit Methods

- Slides
  - https://www.cs.cmu.edu/~baraff/sigcourse/slidese.pdf

- Lecture Notes
  - https://www.cs.cmu.edu/~baraff/sigcourse/notese.pdf

Popular Integration Method 2
- Semi-Implicit Euler Method

The implicit Euler method

The implicit method, for comparison, finds a new position whose derivative can update the current value to the new value (Eq. B.153).

$$y_{n+1} = y_n + hf'(x_{n+1}, y_{n+1}) \tag{B.153}$$

This requires solving an equation to find such a y_{n+1} using, for example, a Newton-Raphson method. It takes more time to take a step using the implicit Euler method but because this technique is more stable than explicit Euler, large time steps can be taken and, thus, an overall computational savings can often be achieved.

The semi-implicit Euler method

The *semi-implicit Euler method* is derived from the implicit Euler method by approximating the y_{n+1} term used in evaluating the derivative by an explicit Euler step, thus resulting in Equation B.154.

$$y_{n+1} = y_n + hf'(x_{n+1}, y_n + hf'(x_n, y_n)) \tag{B.154}$$

The semi-implicit Euler method offers more stability than explicit Euler and is computationally much less expensive than implicit Euler.

Physics Simulation Steps

Physics Simulation Steps

- Physics simulation (Forward dynamics simulation)
  - Force computation
    - Compute force (f) at each time instance
      - Gravitational force, spring force, (ground) contact force, viscous drag force, joint torques, muscle contraction force, ...
  - Forward dynamics
    - Given force (f), compute resulting acceleration (a)
      - Particle dynamics: F = ma
    - "Derivative evaluation" for solving the differential equation
  - Time integration
    - Numerically solving the differential equation (the equation of motion)
      - Euler / Midpoint / Runge-Kutta 4th order (RK4) / Implicit Euler / Semi-implicit Euler integrations ...
    - Object state is update by integration with derivative evaluations.

Physics Simulation Steps

- Physics simulation (Forward dynamics simulation)
  - Force computation
    - Compute force (f) at each time instance
      - Gravitational force, spring force, ground contact force, viscous drag force, joint torques, muscle contraction force, ...
  - Forward dynamics
    - Given force (f), compute resulting acceleration (a)
      - Particle dynamics: F = ma
      - **Rigid body dynamics**
      - **Articulated body dynamics**
    - "Derivative evaluation" for solving the differential equation
  - Time integration
    - Numerically solving the differential equation (the equation of motion)
      - Euler / Midpoint / Runge-Kutta 4^{th} order (RK4) / Implicit Euler / Semi-implicit Euler integrations ...
    - Object state is update by integration with derivative evaluations.

Rendering Simulated Particle System in Your Viewer

- euler_step() 혹은 다른 integration step (simulation step) 함수를 통해 시뮬레이션이 한 timestep씩 진행될텐데, 이 때 매 simulation timestep마다 렌더링을 하지 말 것.
  - 그렇게 하면 realtime performance가 안 나옴. 파티클이 너무 느리게 움직이고 있는 것으로 보일 것임.
  - 사람은 그보다 integration step보다 훨씬 큰 시간간격을 두고 업데이트 되는 이미지도 연속적으로 움직이는 이미지로 인식함.

- 자신의 viewer에서 매 frame 업데이트를 할 때마다 object state가 update되고 rendering 되면 충분함.
  - 현재는 약 1/30초 마다 frame을 업데이트 하는 식일 것임.

- 매 frame 업데이트를 할 때마다 frame time만큼 시뮬레이션을 진행시키고,
  - 예) 1/30초마다 frame이 증가하고 simulation timestep이 1/900초이면, frame 업데이트를 할 때마다 simulation step을 30번 진행.
  - callback 함수 등의 구조 이용.
- Paint event handler에서는 particle system의 현재 state를 얻어와서 그대로 그리면 됨.

Physics-Based Character Control Problem

Controller

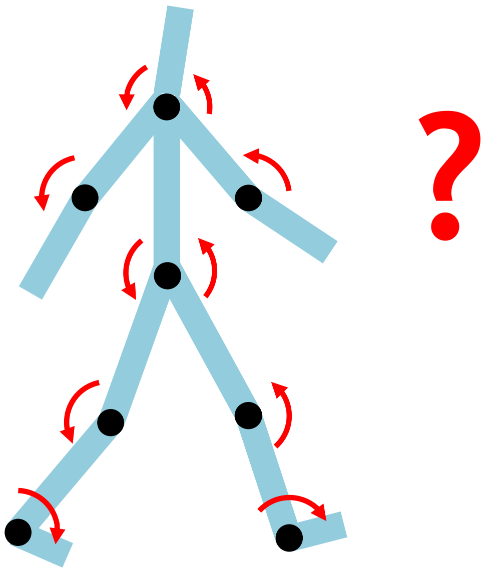
- Let's say we have a physics simulation engine supporting articulated body dynamics.

https://youtu.be/lBNRlrsSGr8

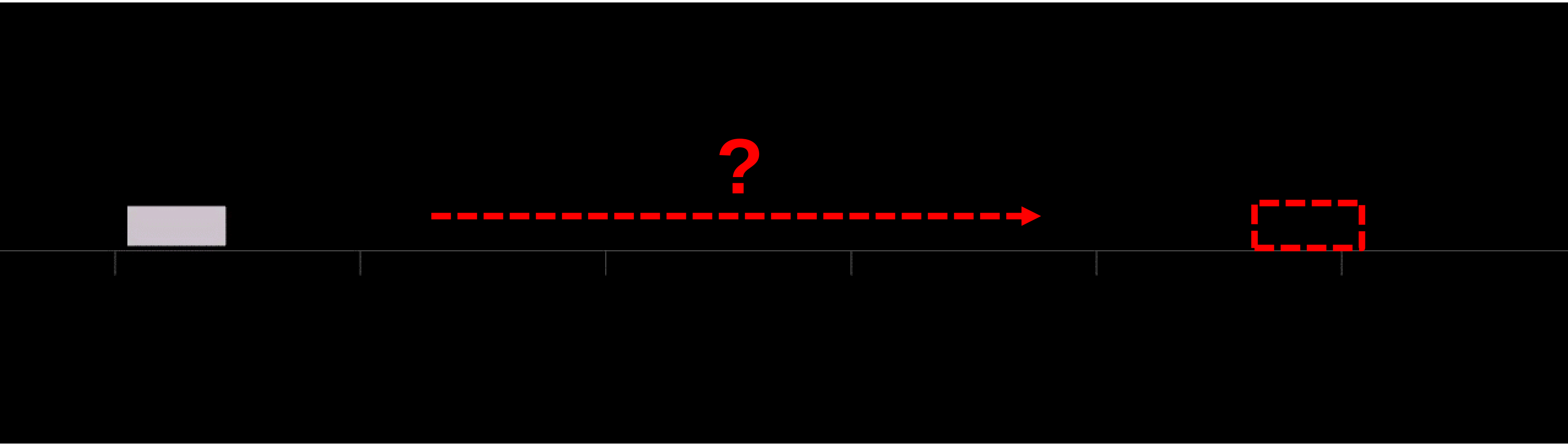
- How can we prevent the character from falling?
- → Set proper joint torques at each timestep based on character's current states
- → What "Controller" (or control algorithm) does.

Controller

- A controller is a device which monitors and affects the operational conditions of a given dynamical system.

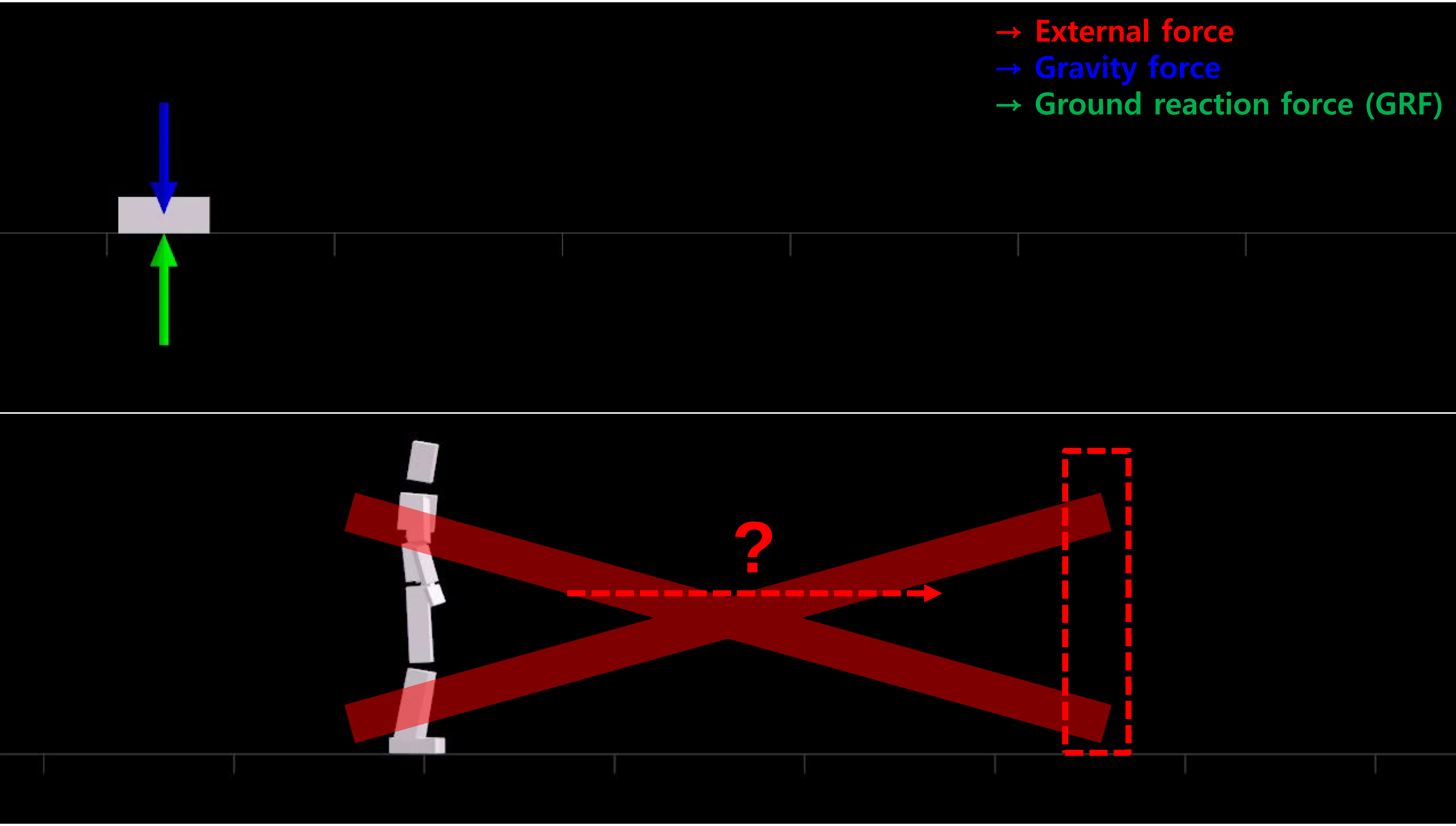


Why is Character Control Challenging?



Why is Character Control Challenging?

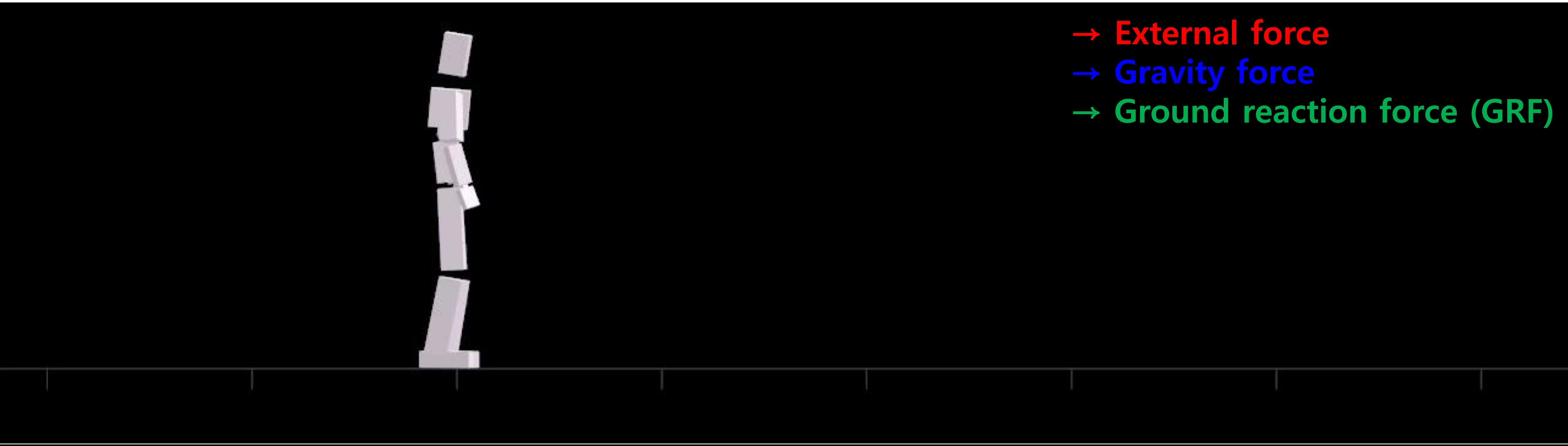
https://youtu.be/oh5YssSMb_o

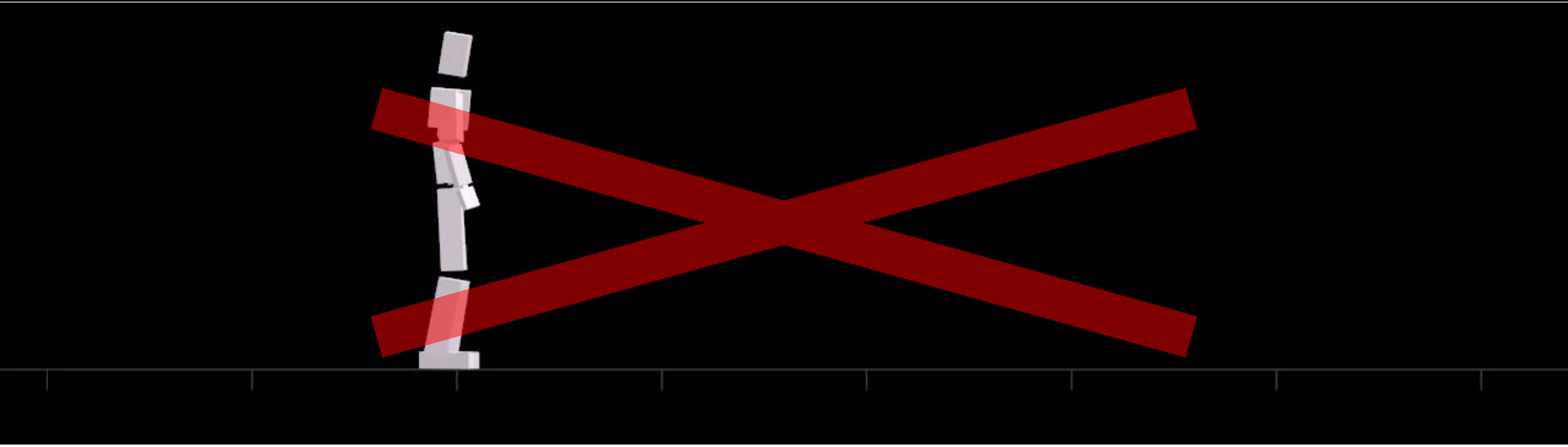


https://youtu.be/QvPcX9hnw0k

Why is Character Control Challenging?

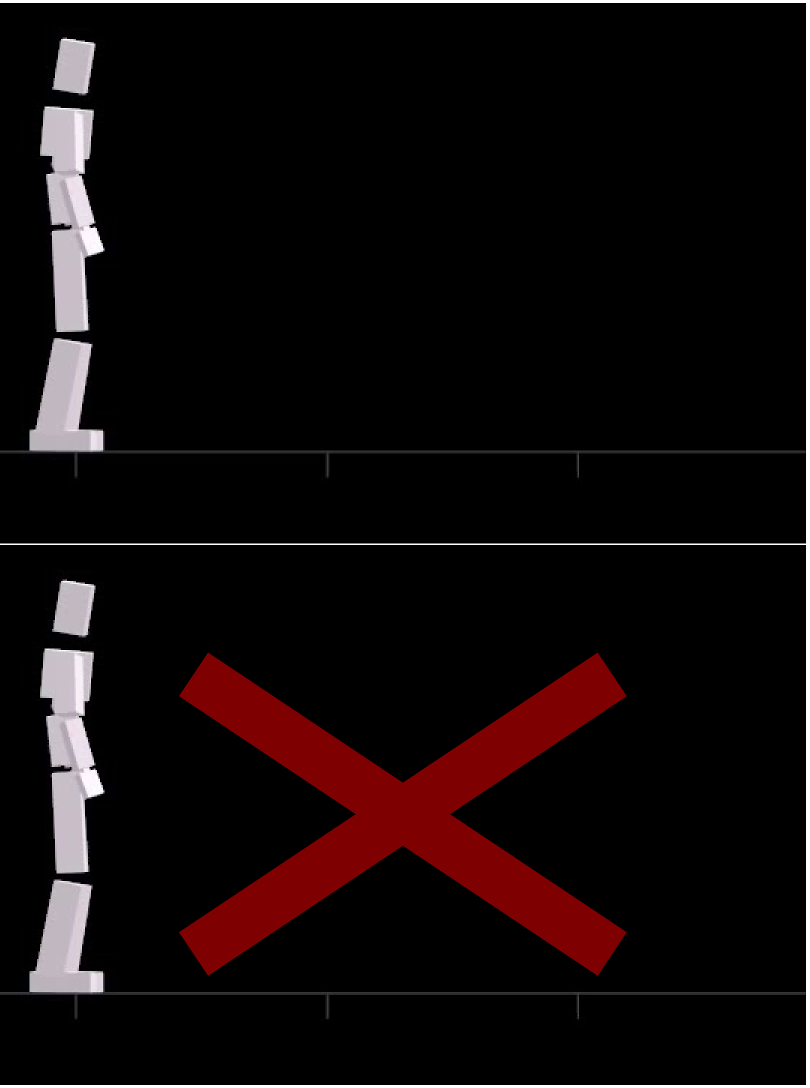
https://youtu.be/yBUQ9inbres



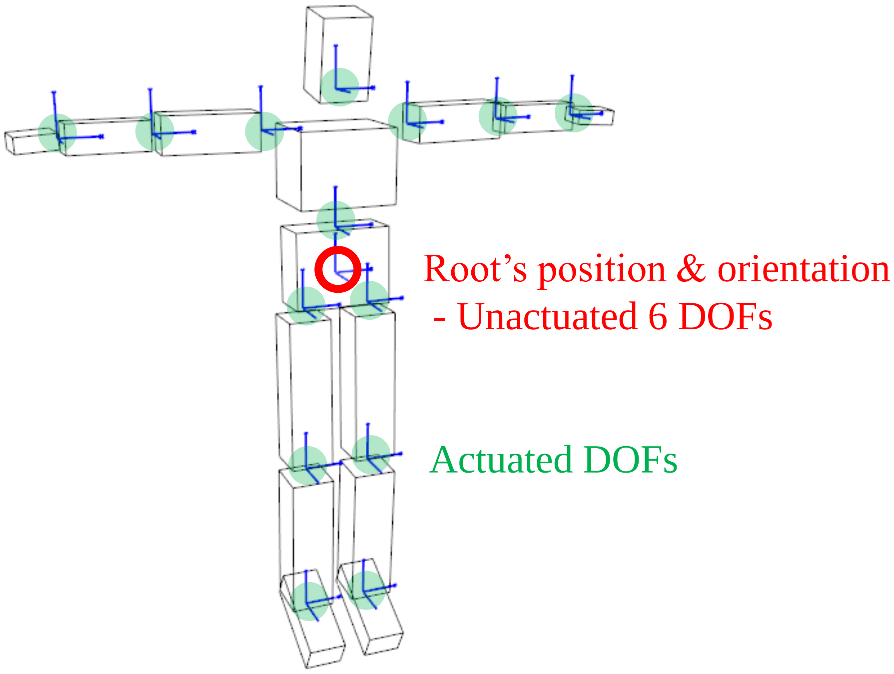


https://youtu.be/QvPcX9hnw0k

Why is Character Control Challenging?

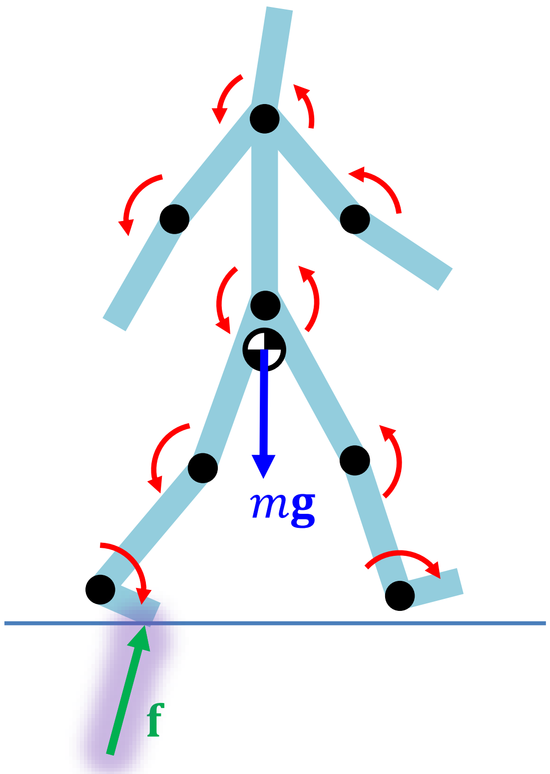


Underactuated System

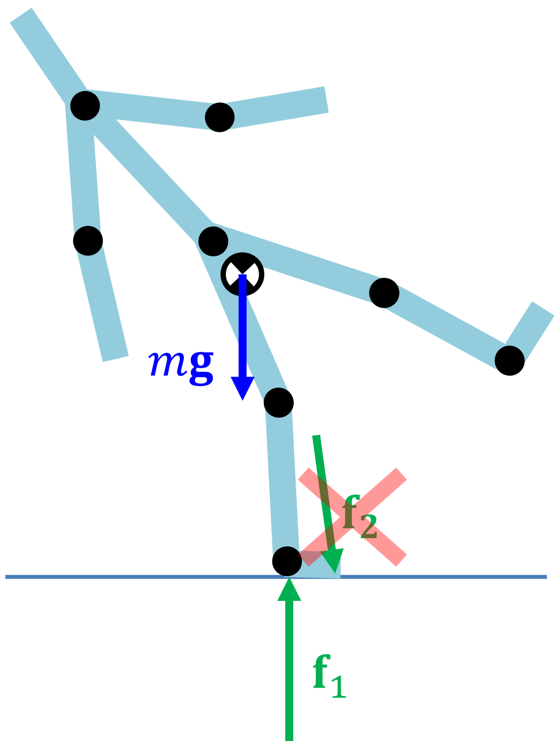


Why is Character Control Challenging?

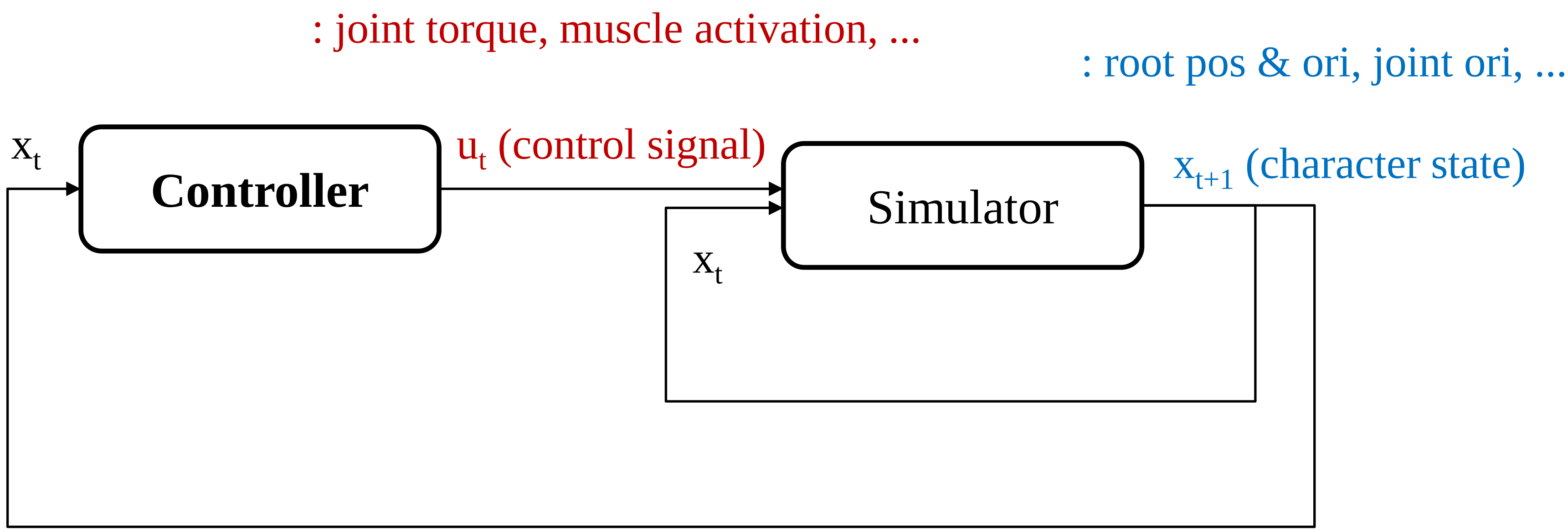
GRF is **indirectly** controlled via internal joint actuation only.



GRF **cannot pull the feet!**



Simulation / Control Loop



Intro to DRL-Based Physics-Based Character Control

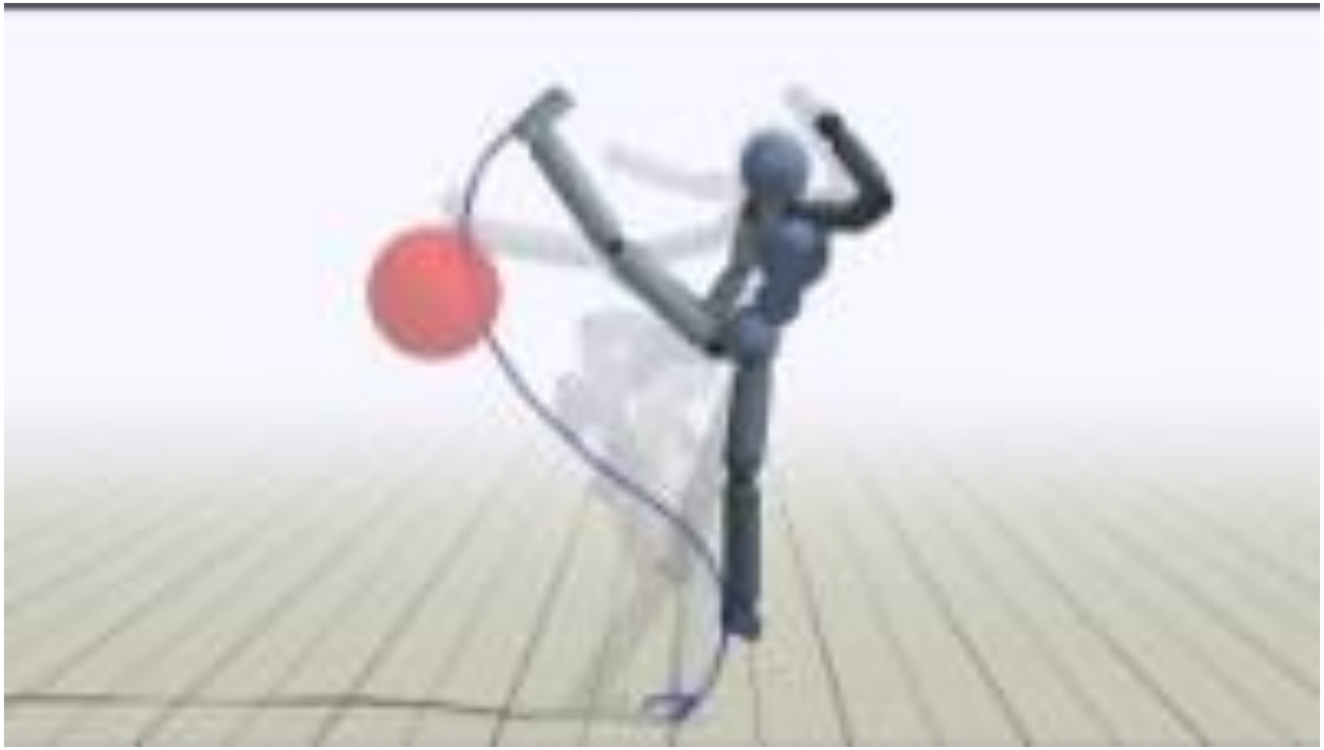
Deep Reinforcement Learning-Based

- 이 문제를 다루는 최근 대부분의 연구들은
  - Deep Reinforcement Learning (DRL)을 사용하여 control policy를 학습함으로써
  - 캐릭터가 물리적으로 타당하고 자연스러운 모션으로 원하는 동작을 수행하도록 제어하는 방식으로 이루어 짐

Problems

- Tracking reference motions
  - [Liu 2017], [Peng 2017], [Liu 2018], [Peng 2018]

https://youtu.be/vppFvq2quQ0



[Peng 2018]

Problems

- Character control with various motions
  - [Bergamin 2019], [Park 2019], [Peng 2019], [Luo 2020], [Merel 2020], [Peng 2020]

https://youtu.be/wySUxZN_KbM

[Peng 2020]

Problems

- Learning without reference motions
  - [Peng 2016], [Yu 2018]

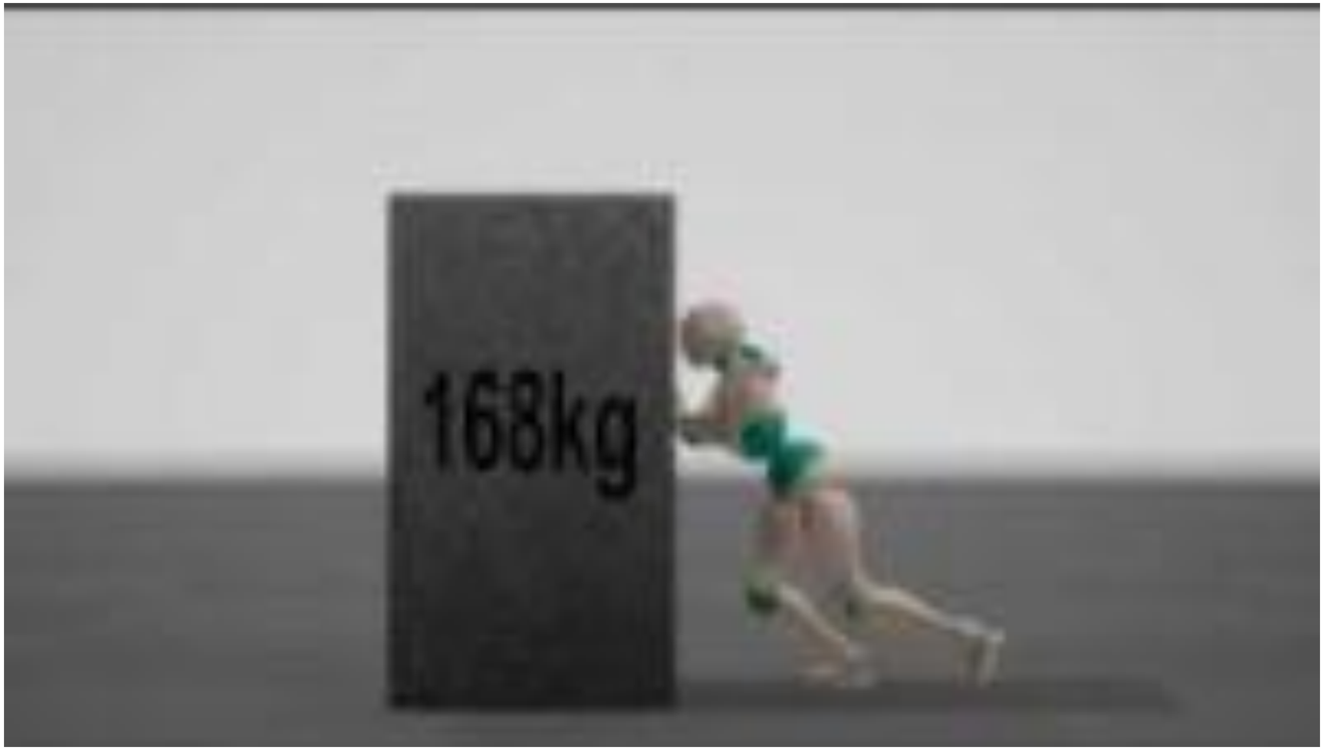
https://youtu.be/zkH90rU-uew

[Yu 2018]

Problems

- Learning policies for motion variations
  - [Won 2020], [Lee 2021], [Yin 2021]

https://youtu.be/ihHttL-SeL4



[Lee 2021]

Problems

- Control of musculoskeletal models
  - [Jiang 2019], [Lee 2019]

https://youtu.be/a3jfyJ9JVeM

[Lee 2019]

Problems

- Control of non-human character
  - [Peng 2016], [Won 2017], [Won 2018], [Luo 2020]

https://youtu.be/t9CdF_Pl19Q

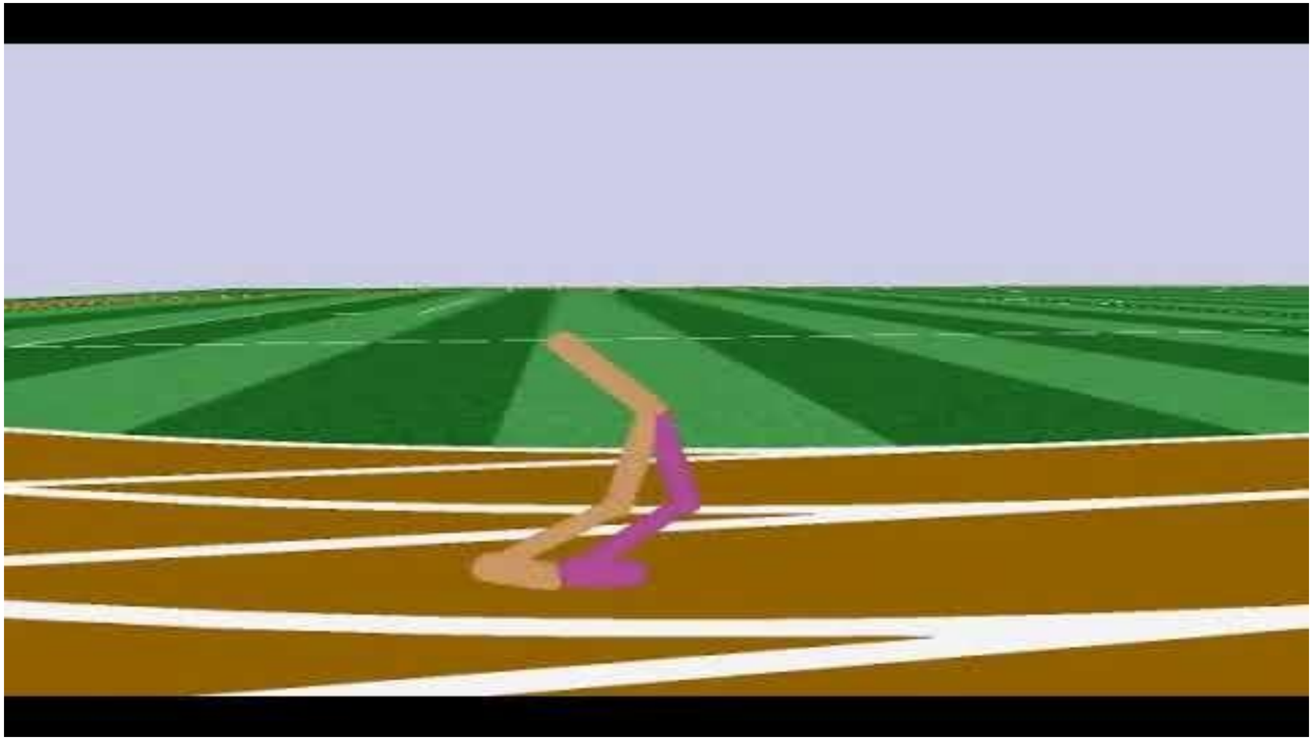
[Luo 2020]

Reinforcement Learning

- Reinforcement learning (RL) is an area of machine learning concerned with how software **agents** ought to take **actions** in an **environment** in order **to maximize** the notion of **cumulative reward.** [Wikipedia]

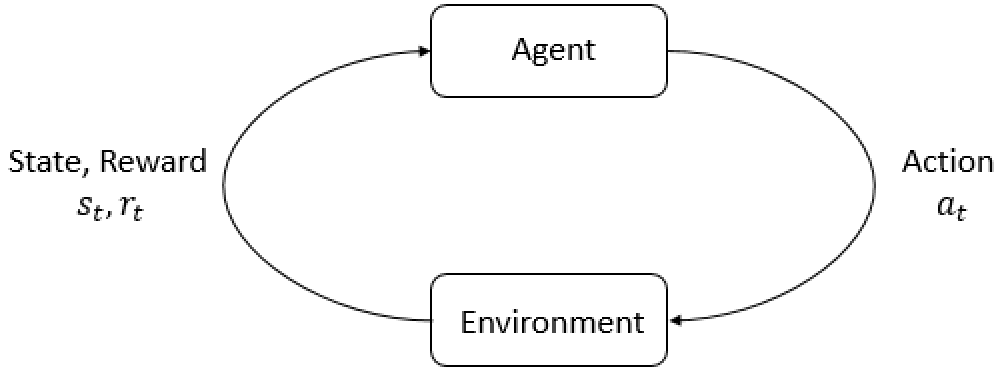
- Much of the following slides come from:
  - https://spinningup.openai.com/en/latest/spinningup/rl_intro.html

RL Environment Example



https://youtu.be/qmfJQRleo5A

Key Concepts



- The **environment** is the world that the **agent** lives in and interacts with.
  - e.g. environment: the track, the 2d walker
  - e.g. agent: the "brain" of the 2d walker

- **At every step** of interaction, the agent **sees** a (possibly partial) **observation** of the **state** of the world, and then **decides** on an **action** to take.
- **Then** the world **state is updated**.
  - e.g. state: joint angles & velocities of the 2d walker
  - e.g. action: joint torques (or joint target angles) to be applied
  - e.g. step update: simulation of 1/30 s

Key Concepts

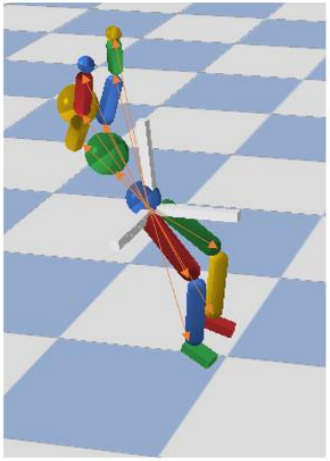
- The agent also perceives a **reward** signal from the environment, a number that tells it *how good or bad* the current state / action taken / next state is.
  - e.g. reward: 2d walker moving distance or speed, distance from the goal position, similarity to the reference motion, or not falling down, ...

- The **goal** of the agent is to **maximize its cumulative reward**, called ***return***.

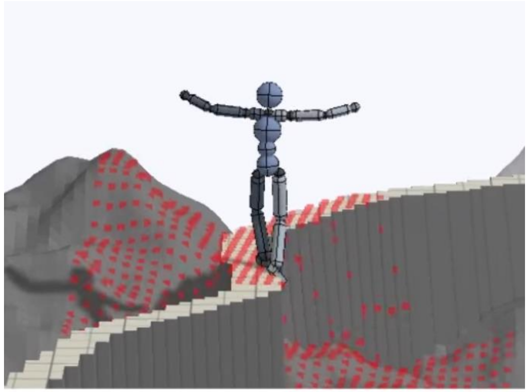
- *Reinforcement learning methods are ways that the agent can learn behaviors to achieve its goal.*

Input for Policy

- Input (state):
  - Character state: often includes
    - position
    - velocity
    - orientation
    - angular velocity
    - of each link or joints, relative to root, w.r.t. root frame or horizontally-aligned root frame



  - Environmental info (if needed)
    - e.g. height map



Output for Policy

- Output (action):
  - Joint torques or
  - PD target pose or
    - Known to be more stable and faster in RL
  - Muscle activations
    - For musculoskeletal models
- All of these are continuous action space.

Rewards

- The weighted sum of reward terms are commonly used.

- Reward terms can be:
  - Imitation reward: penalize difference between current pose of reference motion and current character pose
  - Upright reward: penalize difference between the vector from pelvis to neck and the global vertical axis
  - Energy reward: penalize magnitude of torques
  - Task reward: penalize difference between ...
    - the goal position and the current character position
    - the goal velocity and the current character velocity
    - ...
  - ...

Next Time

- Next two weeks:
  - Preparing Project 4 (No class)

- Dec 9:
  - Presentation: Project 4