ETH zürich

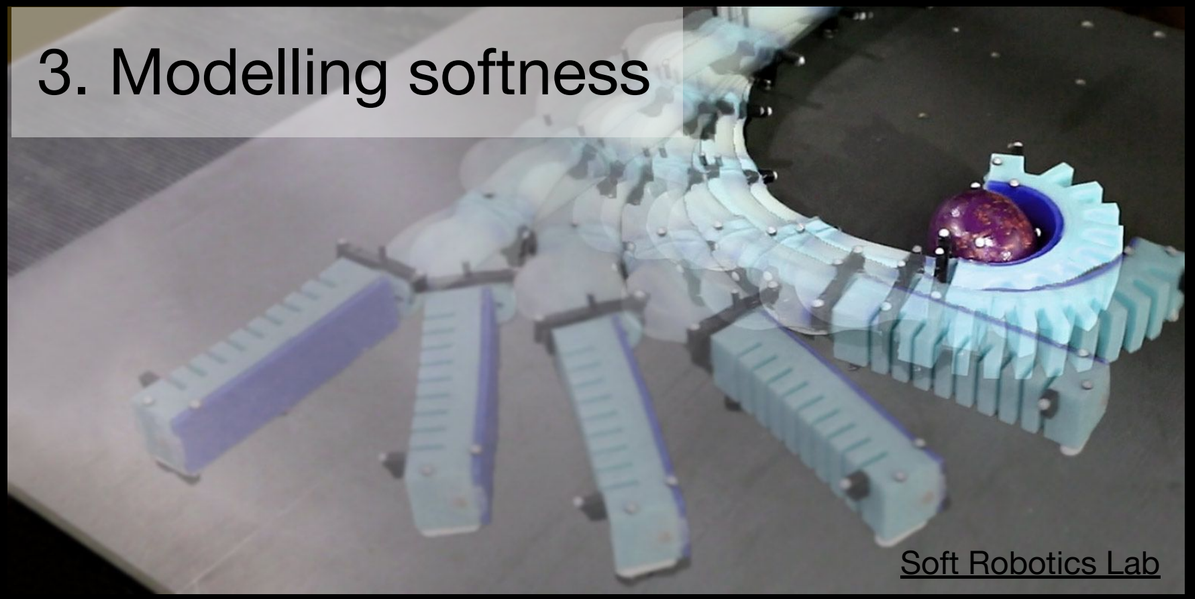
Simulating Robots and Soft Interaction with the World

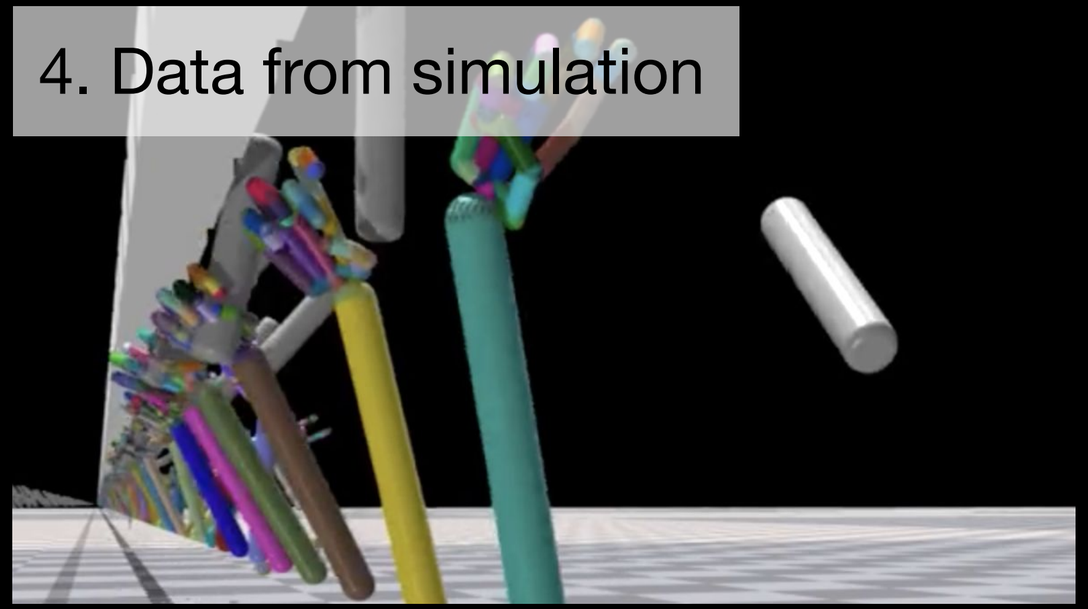
Robert Katzschmann

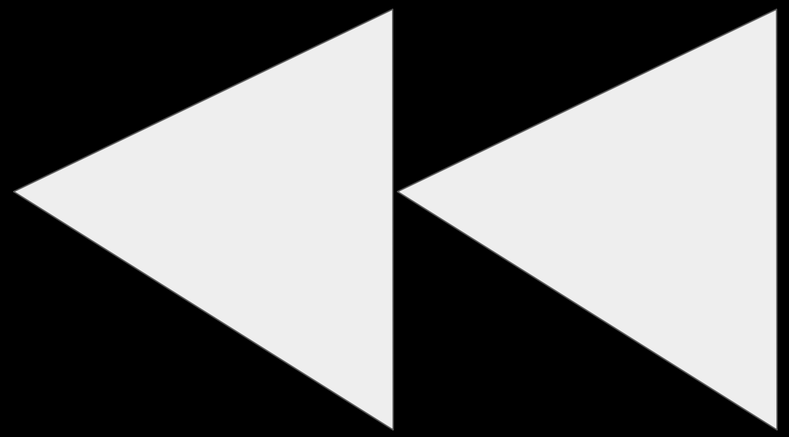
Assistant Professor of Robotics, Soft Robotics Lab

Nvidia Technical Blog

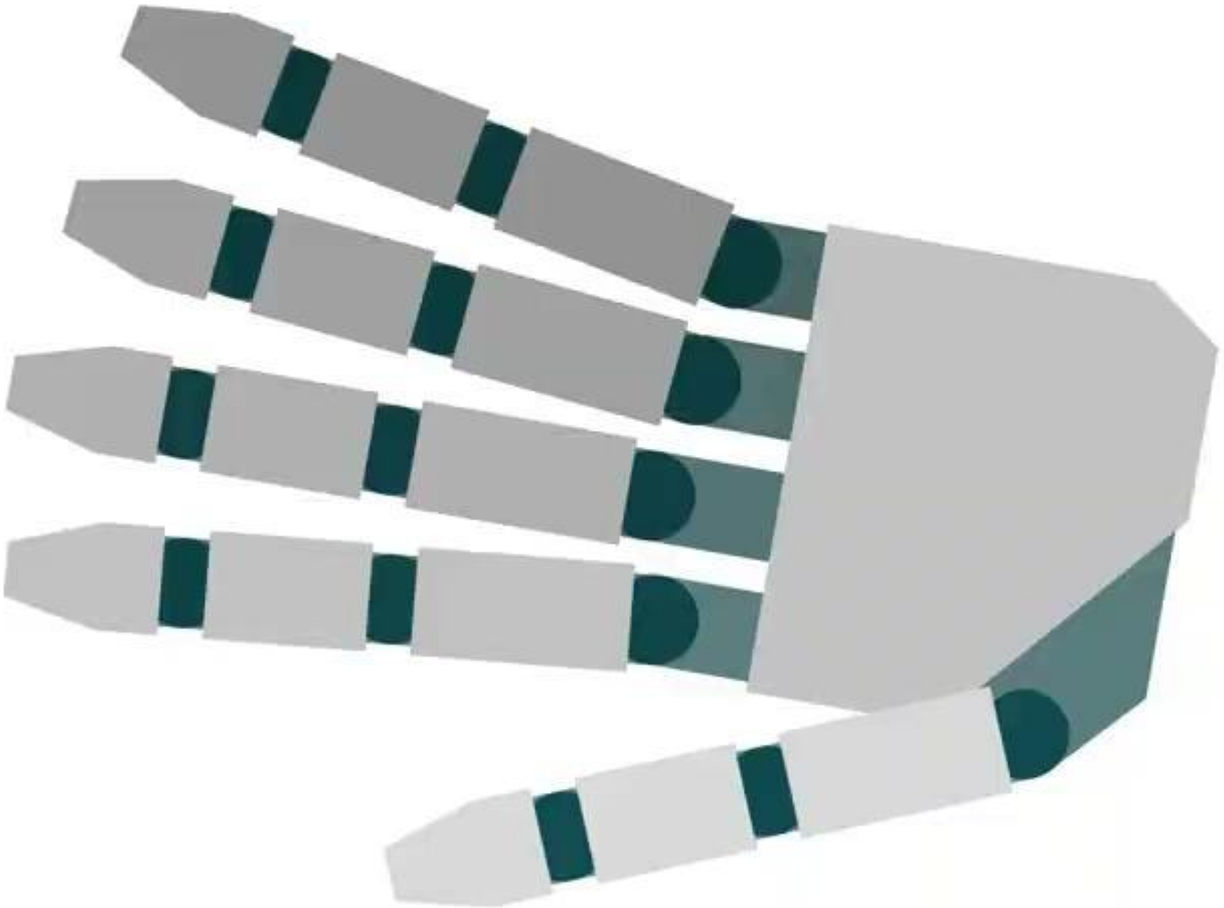
Plan for today





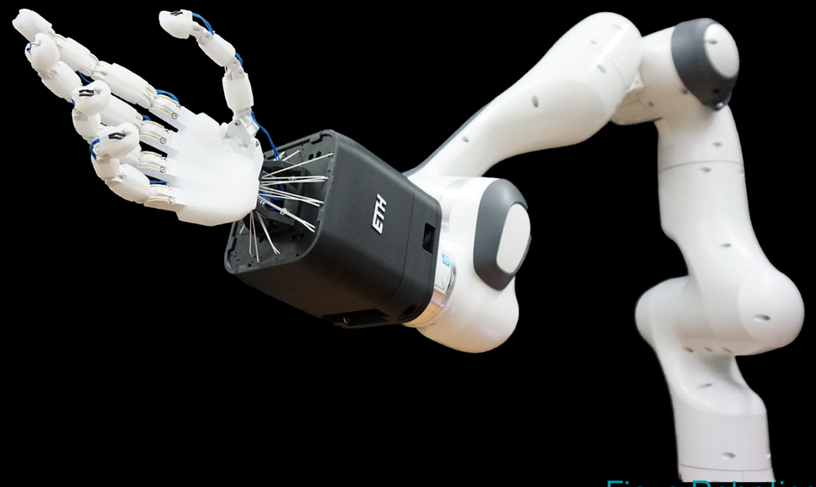


SOFT ROBOTICS - HAND BONES

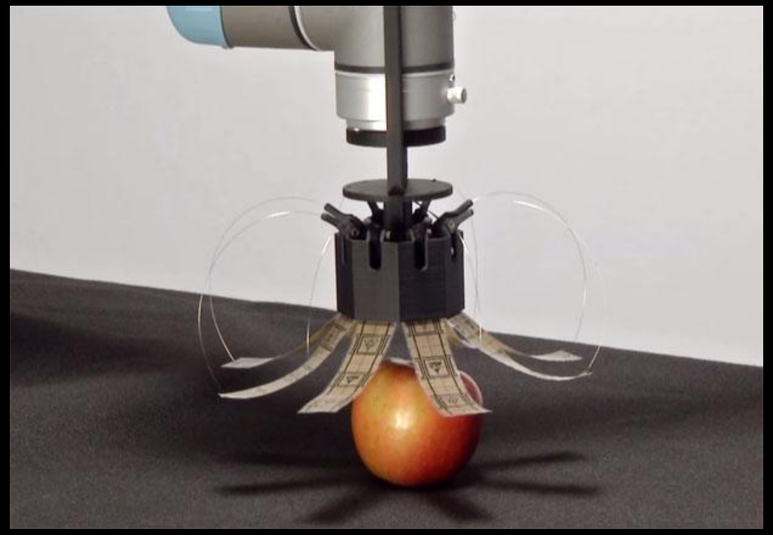


Possible Designs

Robotiq



Fiave Robotics



Source

Each team should design its own personal gripper. There are no limits to your choices. As long as your design is successful in The Challenge …

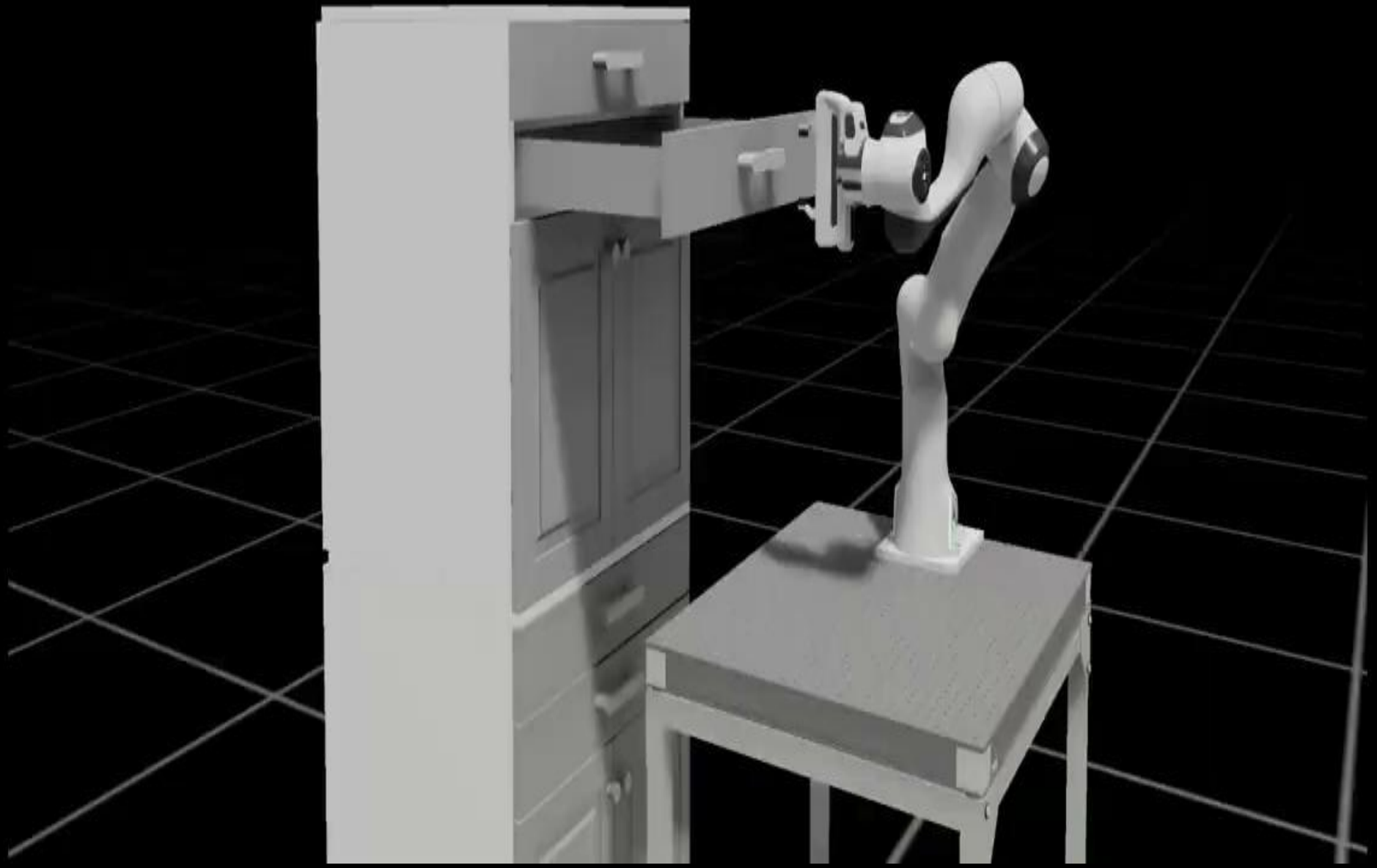
Simulate, but how?

ETH zürich

Part 1: Simulators

Toshimitsu et al., Getting the ball rolling (2023)

SoftRobotics Laboratory

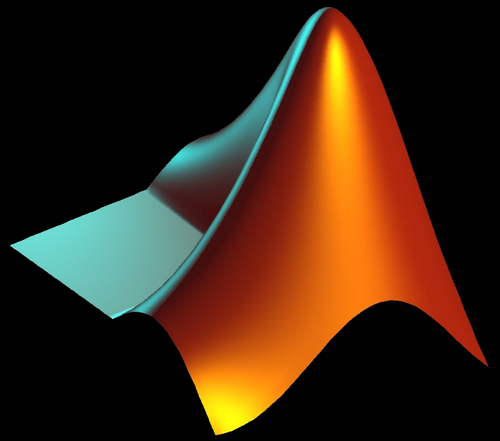


Simulation Frameworks

Sofa

Nvidia Isaac

Mujoco



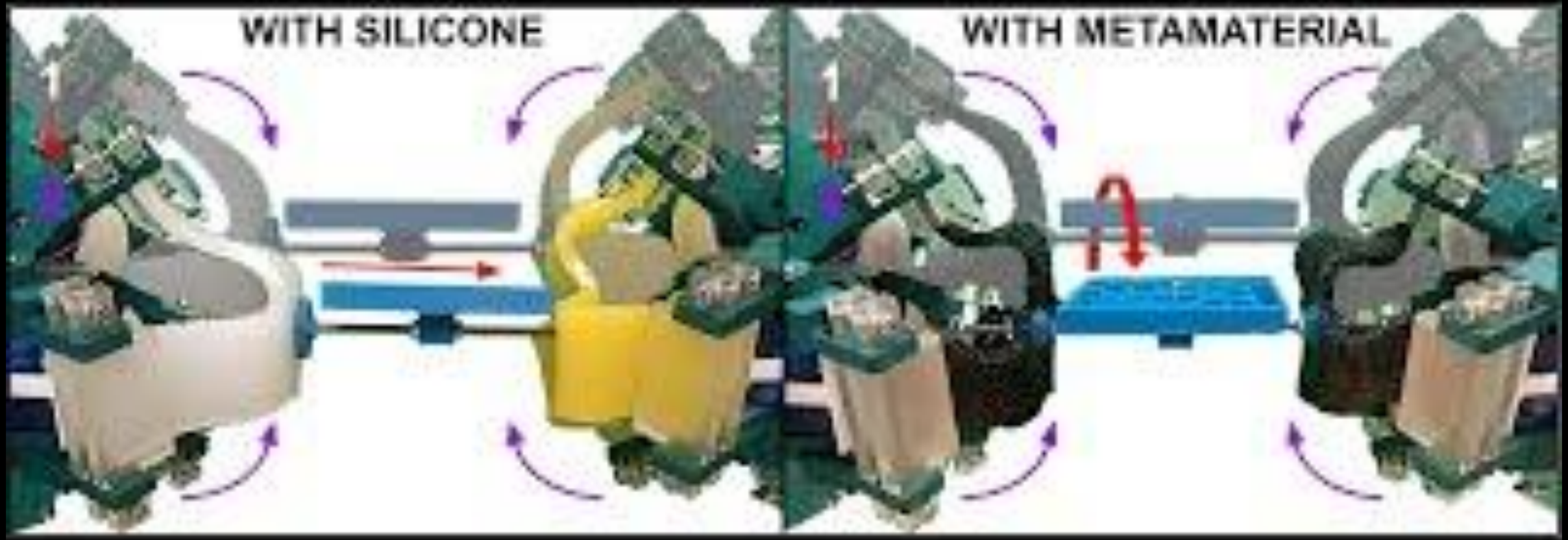
Mathworks

Gazebo

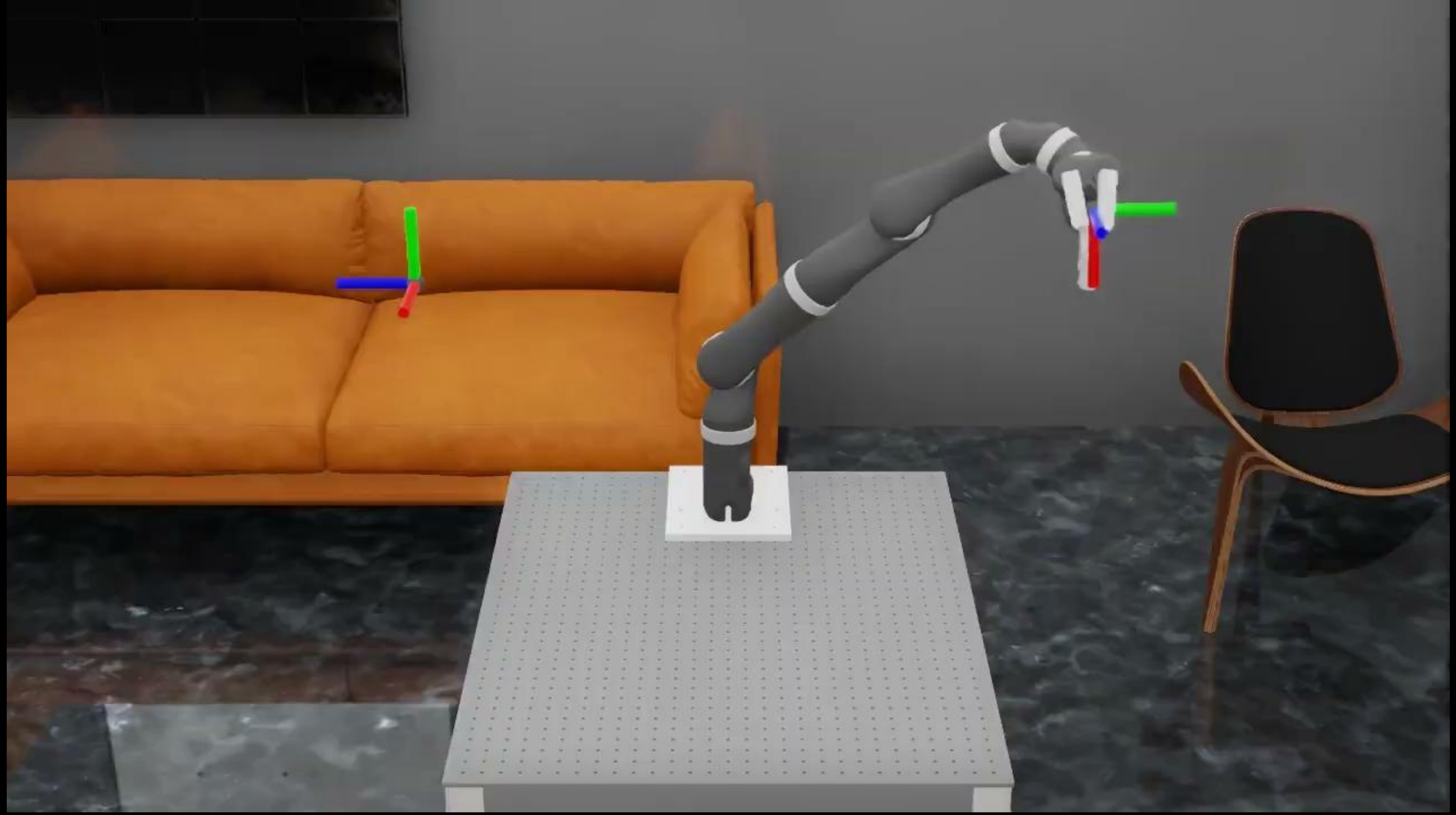
Drake

ETH zürich

SoftRobotics Laboratory

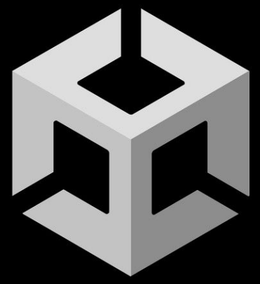


SOFA: Models and Actuators for Soft-Robots

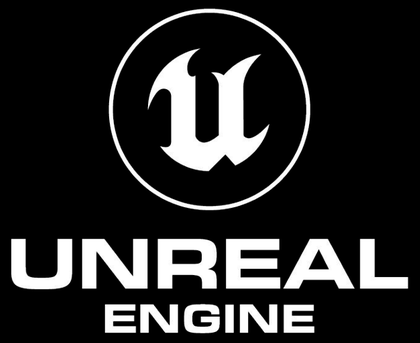


Soft Robotics Lab

Omniverse



Omniverse is a development platform for building and operating custom 3D pipelines and industrial metaverse applications with Universal Scene.



It connects with many features and apps useful to creators and developers.

Between the big players of 3D rendering we can find Isaac Gym.

ETH zürich

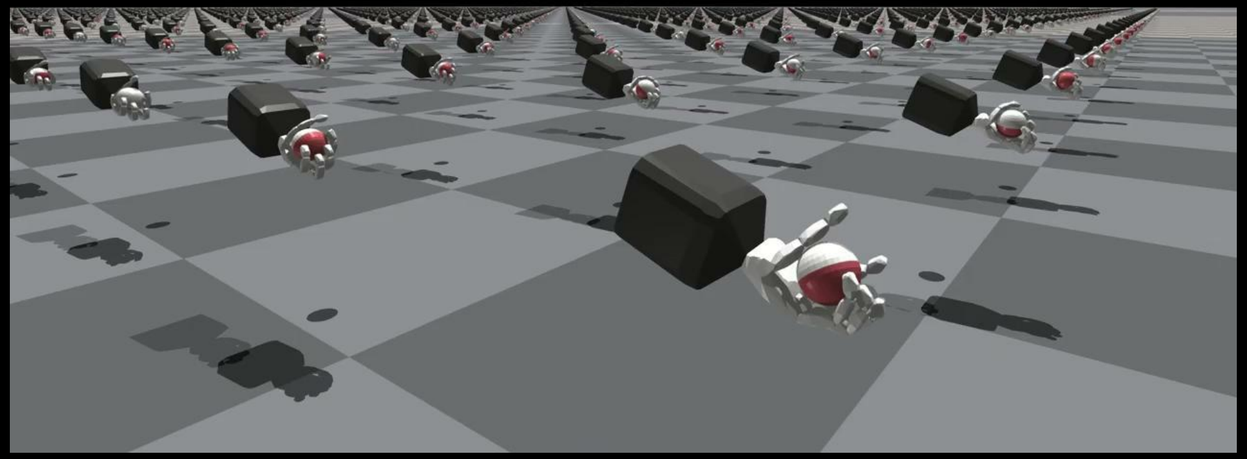
Soft Robotics Laboratory

Isaac Sim

Isaac Sim is a simulation application and synthetic photo-realistic generator tool.

It allows to create a digital twin, generate synthetic data and much more.

Isaac Gym



Toshimitsu et al., Getting the ball rolling (2023)

A physics simulation environment for reinforcement learning research.

Physics simulation in Isaac Gym runs on the GPU, storing results in PyTorch GPU tensors.

Observations and rewards can be calculated on the GPU in PyTorch, enabling thousands of environments to run in parallel on a single workstation.

Isaac Orbit

ORBIT is a unified and modular framework for robotics and robot learning, powered by Isaac Sim. It offers a modular design to create robotic environments with photo-realistic scenes, and fast rigid and soft body simulation.

It aims to support various research areas, including representation learning, reinforcement learning, imitation learning, and motion planning.

Isaac Orbit Repository and Paper

MuJoCo

Kevin Zakka

Multi-Joint dynamics with Contact

- Unified continuous-time formulation of constraints via convex optimization
- Constraints include soft contacts, limits, dry friction, equality constraints
- Actuators including motors, cylinders, muscles, tendons, slider-cranks
- XML model format (called MJCF) and built-in model compiler
- Cross-platform GUI with interactive 3D visualization in OpenGL
- CPU-based framework

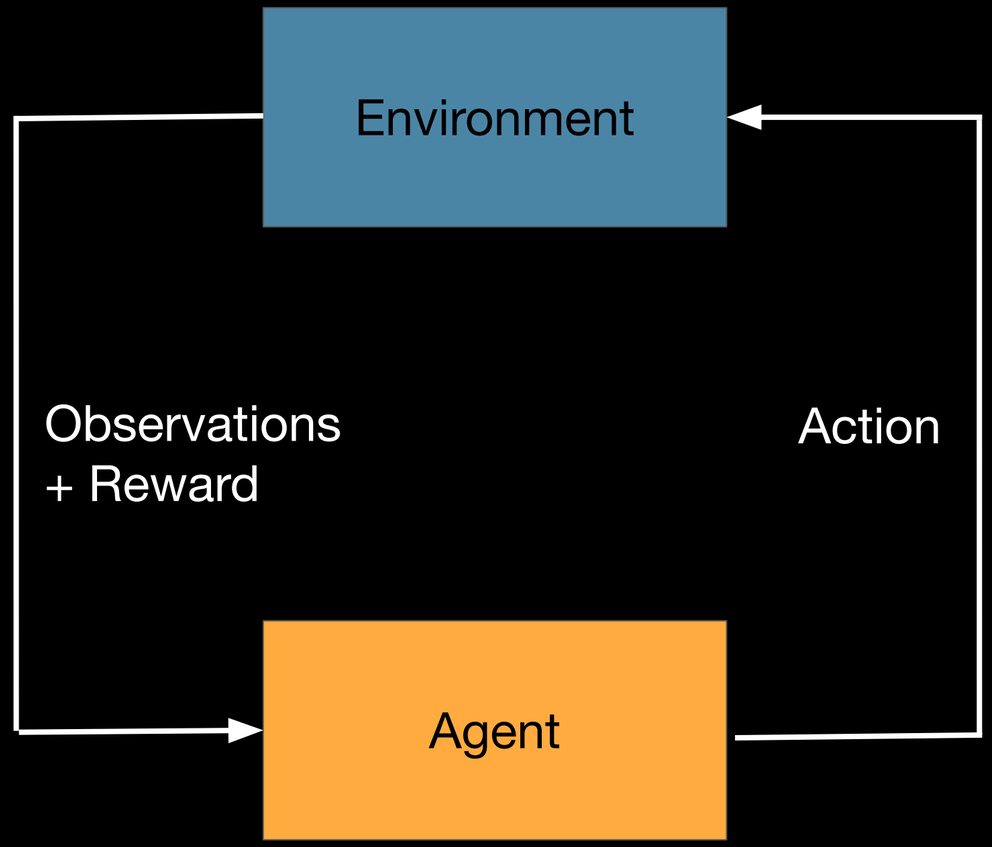
To recap

- Sofa is a stand-alone framework specialized in soft body simulation
- Isaac Gym is used for reinforcement learning training
- Orbit is the next generation of simulators from Nvidia with broader machine learning possibilities
- MuJoCo is a physics engine widely used for machine learning, testing control algorithms or motion planning

You will all have access to Isaac Gym for the second part of the challenge!

Learning + Simulation?

Reinforcement Learning in Simulators

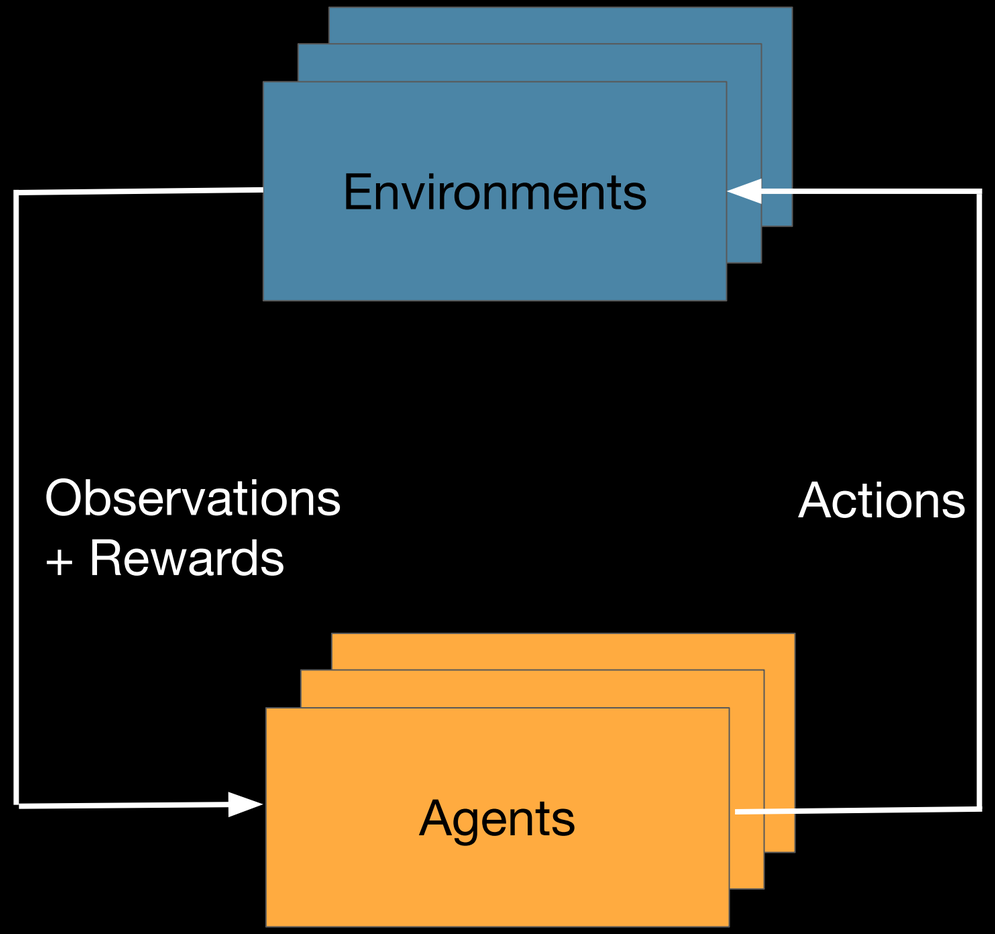


At each time step:

1. The Environment simulates the next states, i.e., position, velocities, acceleration…
2. The observations are passed to the agent
3. The Agent tells the environment what are the next action, i.e. the commands for the actuation of the motors
4. The environments registers the action and the cycle keeps going

Source

Reinforcement Learning in Simulators



In RL you don't just train one agent at the time, you can parallelize!

A neat trick for speed things up is to train thousands of agents simultaneously.

Each one of them runs its own simulation in parallel and independently with the others. For clarity, when you train, you see them all together in the same scenario.

Source

What can we learn from the simulations?

Simulators naturally compute:

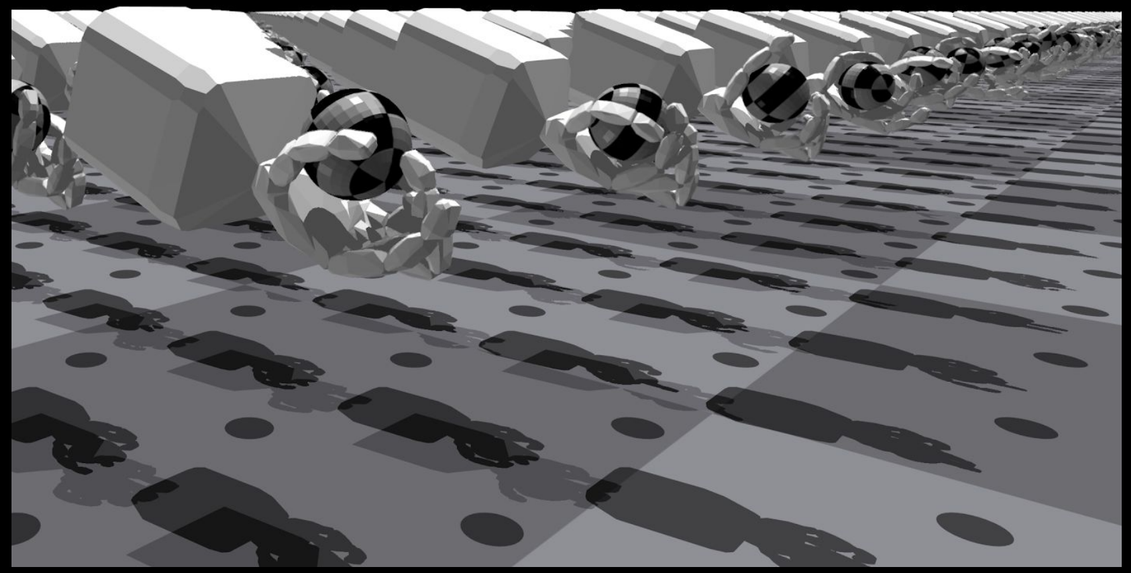
1. Positions
2. Velocities
3. Accelerations
4. Forces at play

WandB:

- create reports on your experiments
- compare trainings
- evaluate performances
- reproduce models

WandB

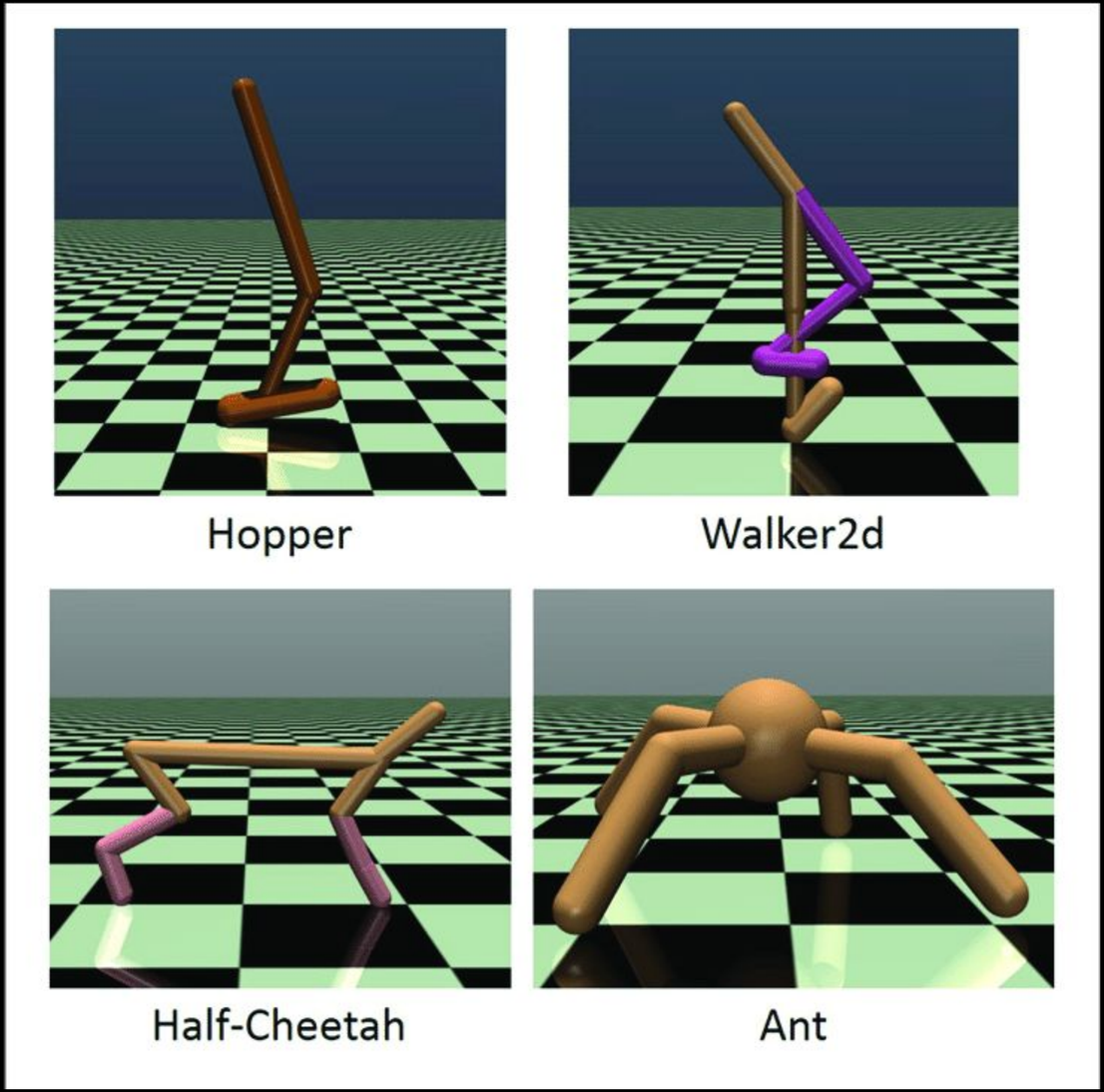
?



ETH zürich

SoftRobotics Laboratory

URDF/MJCF



Eiji Uchibe Cooperative and Competitive Reinforcement and Imitation Learning (2018)

Both are XML-based formats used for robotics.

While URDF is more popular, it comes with more limitations.

For both you will need to implement a hierarchical kinematic tree, specifying connections and dependances in order to have an accurate robotic model.

You will used MuJoCo's Format in the tutorial and during the class.

Katzschmann et al. Exploration of underwater life with an acoustically-controlled soft robotic fish (2018)

Next up…

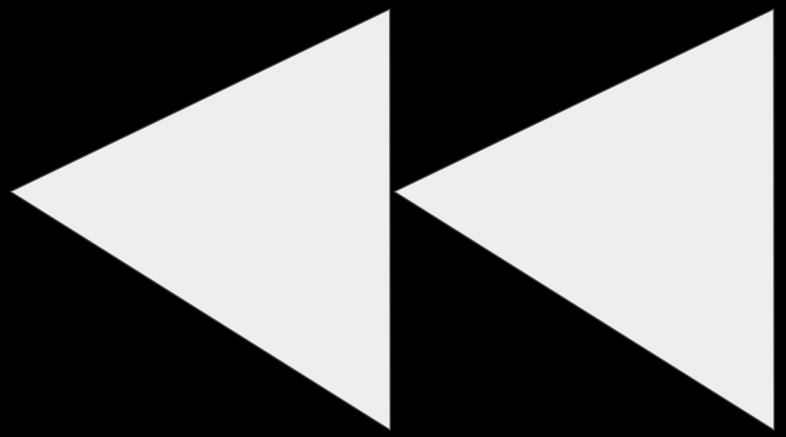
…MODELLING

ETH zürich

NVIDIA Isaac Gym

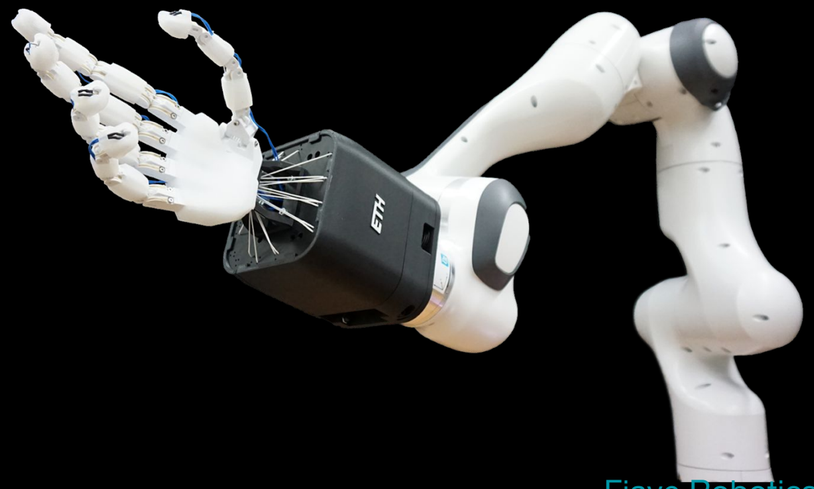
Part 2:
Modelling & MuJoCo

SoftRobotics Laboratory

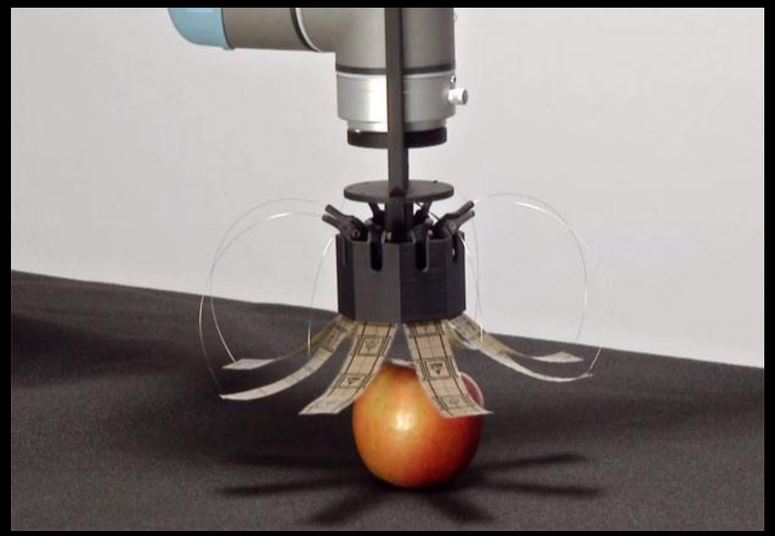


Possible Designs

Robotiq

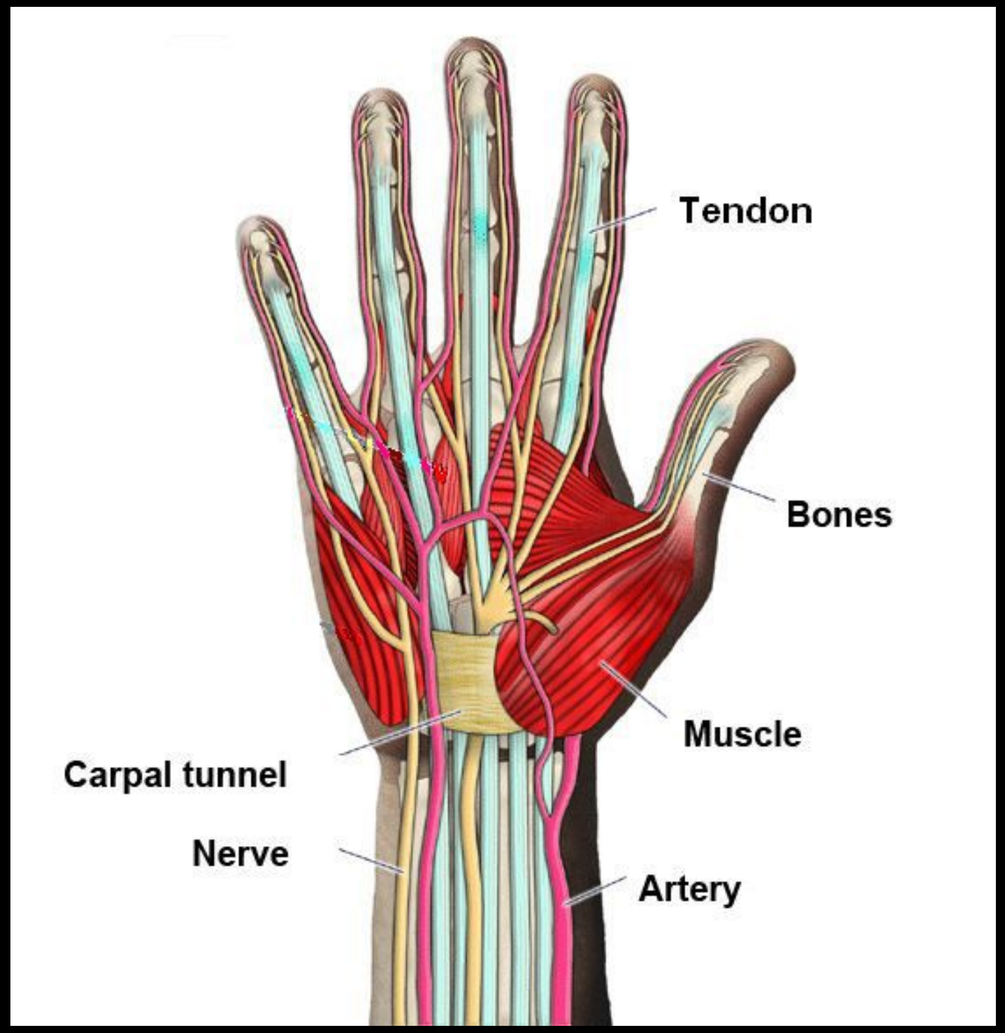


Fiave Robotics

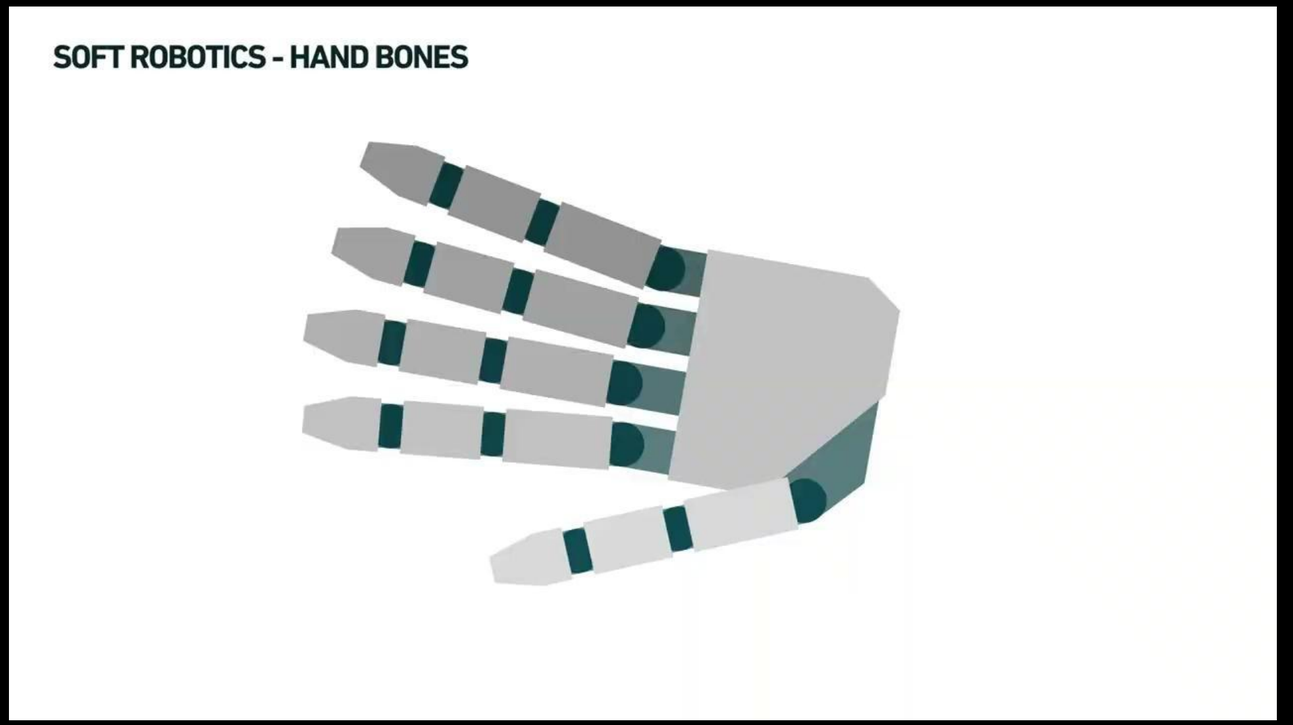


Source

From Reality to a Model



Hand Surgery Group



From Reality to a Model

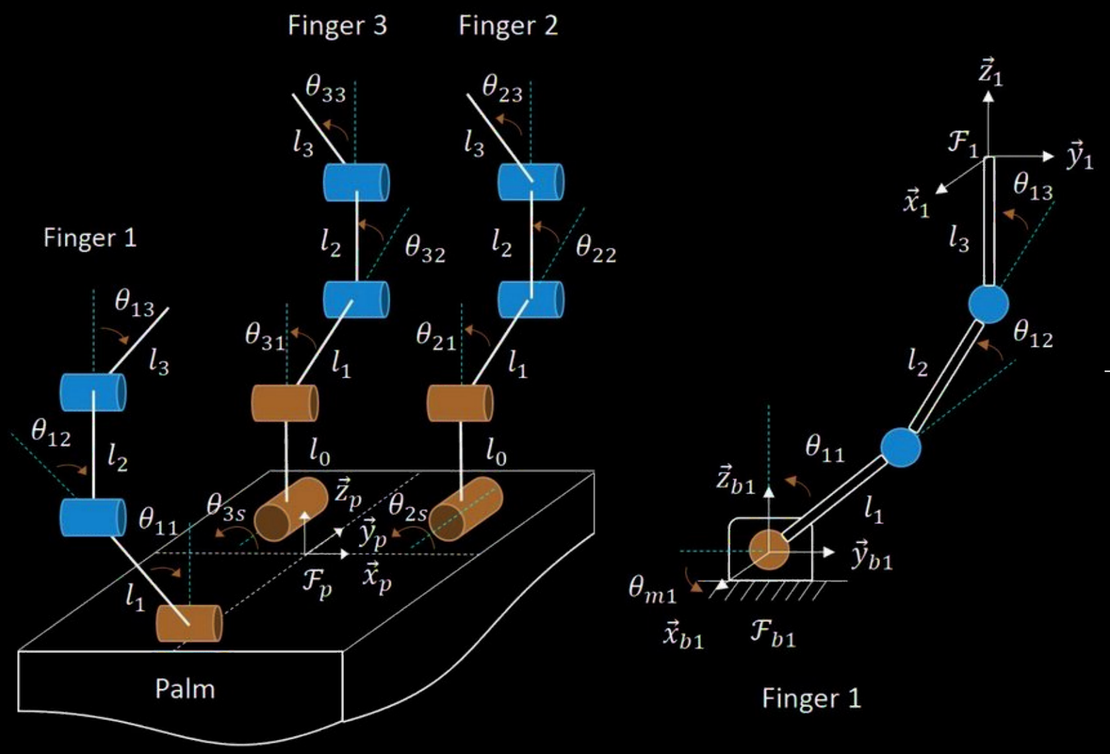
Robotis

We provide you with 11 servo motors for your design.

You also have an additional group budget that you can use if you feel in need of more materials.

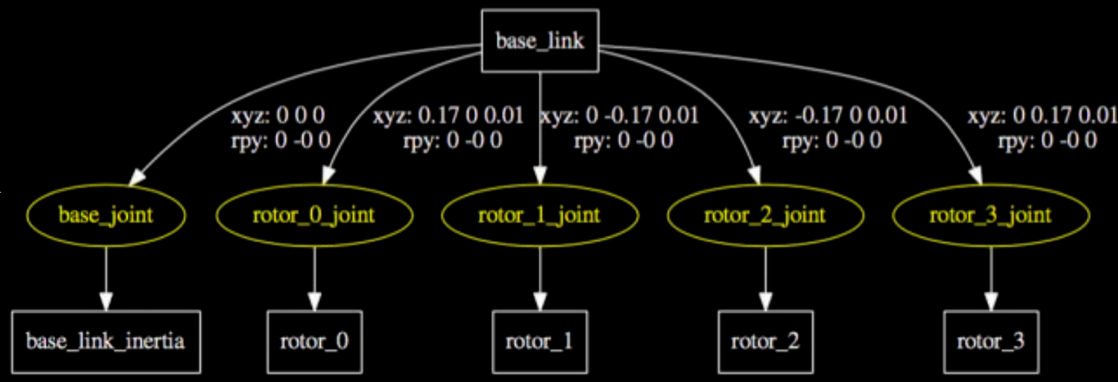
Servo motors can be very fragile, so be cautious.

From Reality to a Model: MuJoCo



Reis et al. Modeling and Control of a Multifingered Robot Hand (2015)

MuJoCo



Will Selby

From Reality to a Model: MuJoCo

1. Always start with
   a. `<mujoco model="ant">`
   b. `Your robot`
   c. `</mujoco>`
2. Inside you can specify more options
   a. `<option timestep="0.016" iterations="50" tolerance="1e-10" solver="Newton" jacobian="dense" cone="pyramidal"/>`
   b. `<visual>`
   c. `<map force="0.1" zfar="30"/>`
   d. `<rgba haze="0.15 0.25 0.35 1"/>`
   e. `<quality shadowsize="2048"/>`
   f. `<global offwidth="800" offheight="800"/>`
   g. `</visual>`

From Reality to a Model: MuJoCo

3. You can also specify meshes
   a. `<asset>`
   b. `<mesh name="robot0:forearm" file="forearm_electric.stl"></mesh>`
   c. `<mesh name="robot0:forearm_cvx" file="forearm_electric_cvx.stl"></mesh>`
   d. `<mesh name="robot0:wrist" scale="0.001 0.001 0.001" file="wrist.stl"></mesh>`
   e. `</asset>`
4. Then the kinematic tree inside
   a. `<worldbody></worldbody>` or
   b. `<body name="root"></body>`

From Reality to a Model: MuJoCo, kinematic tree

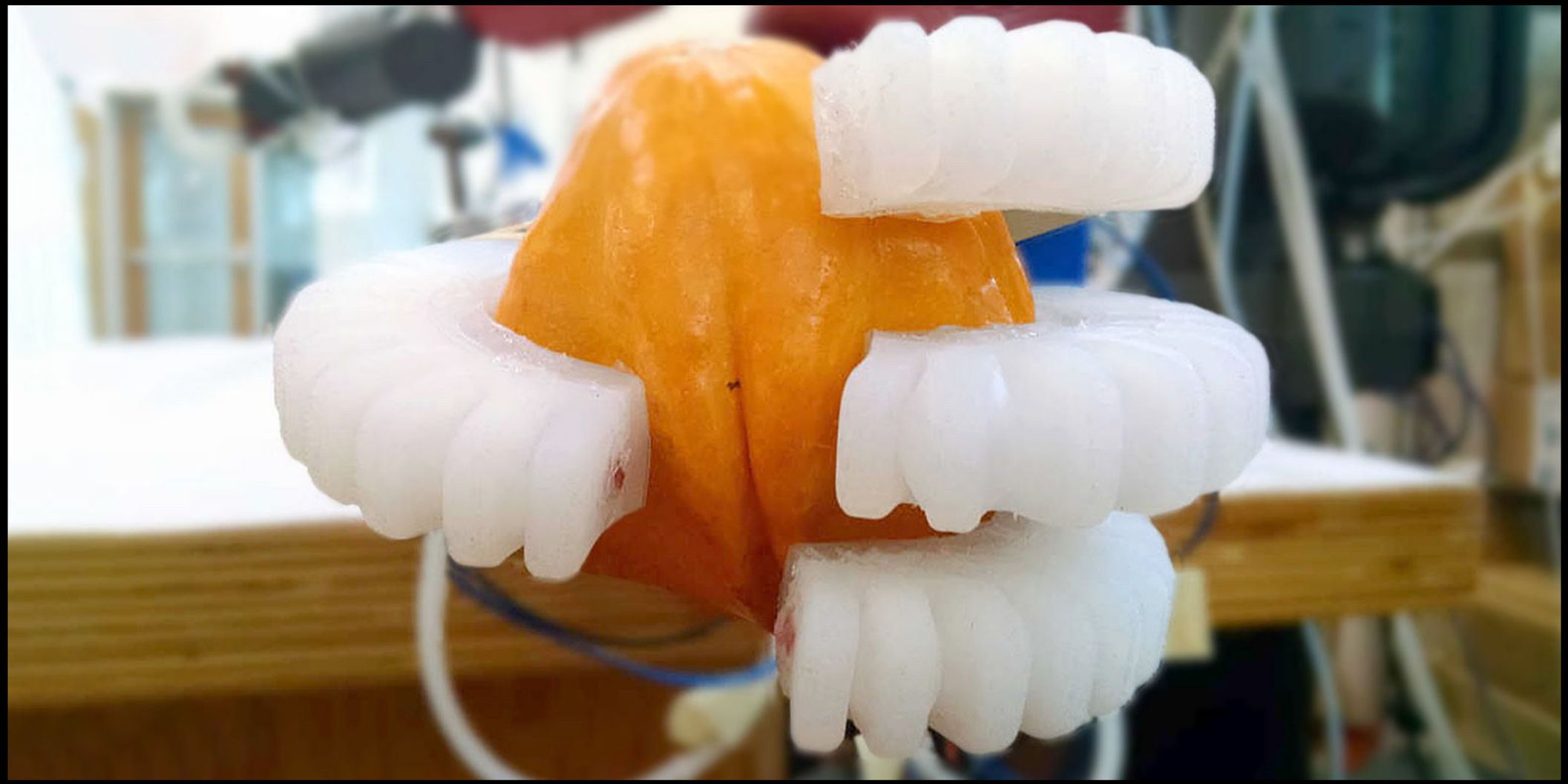
5. A floor
   a. `<geom name="floor" pos="0 0 0" size="0 0 .25" type="plane"/>`
6. A body
   a. `<body name="robot0:hand mount" pos="1 1.25 0.15" euler="1.5708 0 3.14159">`
      i. Euler Angles
      ii. Quaternions
      iii. Axis-angle
      iv. More can be found here
   b. Inertia
      i. `<inertial mass="0.1" pos="0 0 0" diaginertia="0.001 0.001 0.001"></inertial>`
   c. No joints for fixed objects

From Reality to a Model: MuJoCo, kinematic tree

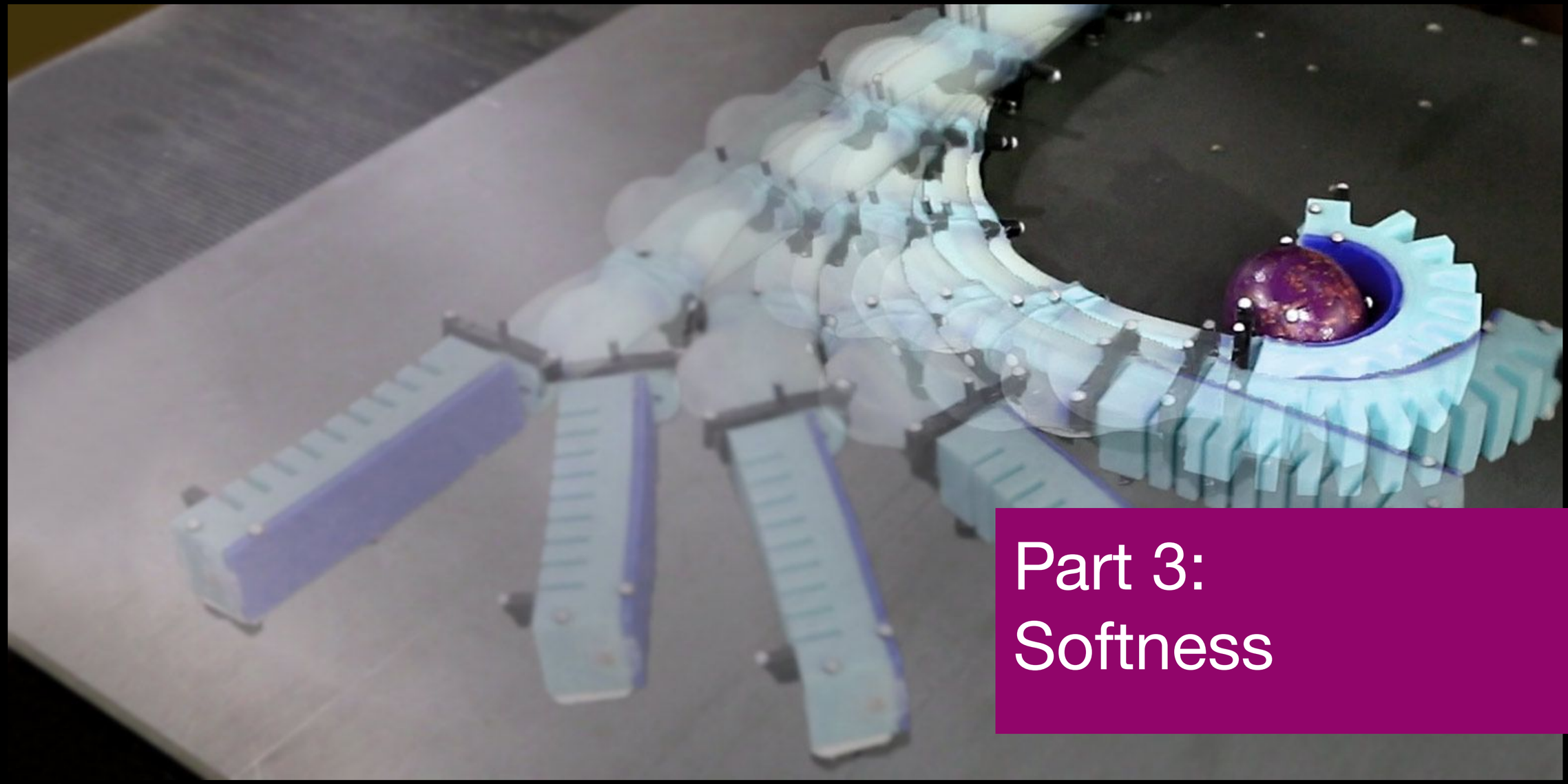
7. Next up the wrist
   a. `<body name="robot0:wrist" pos="0 0 0.256">`
      i. `<inertial pos="0.003 0 0.016" quat="0.504 0.496 0.495 0.504" mass="0.3" diaginertia="0.001 0.001 0.001"></inertial>`
      ii. `<joint name="robot0:WRJ1" type="hinge" pos="0 0 0" axis="0 1 0" range="-0.489 0.14" damping="0.5" armature="0.005" user="1123"></joint>`
      iii. `<geom name="robot0:V_wrist" type="mesh", mesh="robot0:wrist"></geom>`
8. Out of `<worldbody></worldbody>` more informations about the motors can be described
   a. `<actuator>`
      i. `<motor name=”motor1” ctrllimited="true" ctrlrange="-1.0 1.0" joint="robot0:WRJ1"/motor>`
   b. `</actuator>`

XML Reference, GitHub Isaac Envs

What about softness?



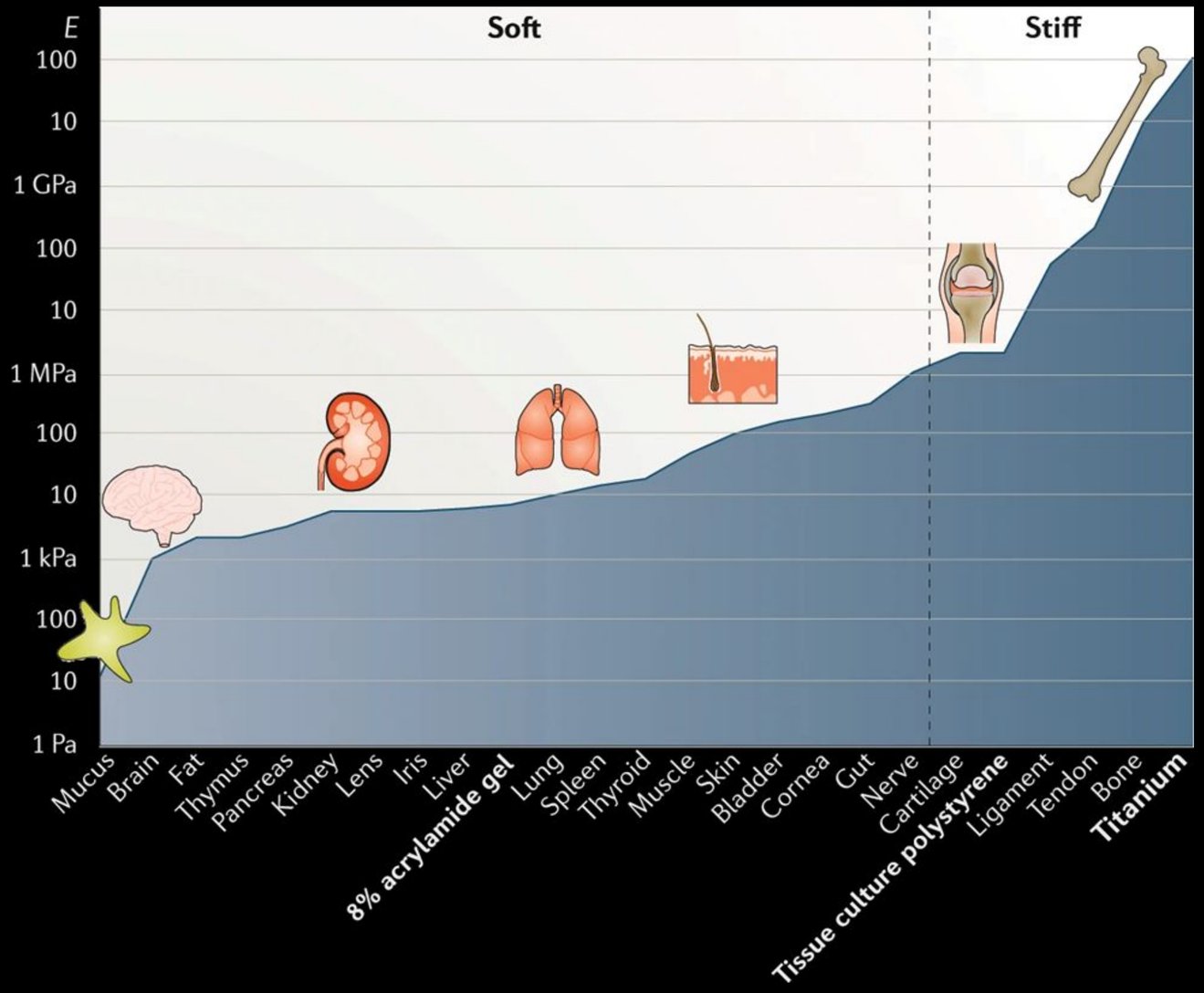
ETH zürich



Part 3: Softness

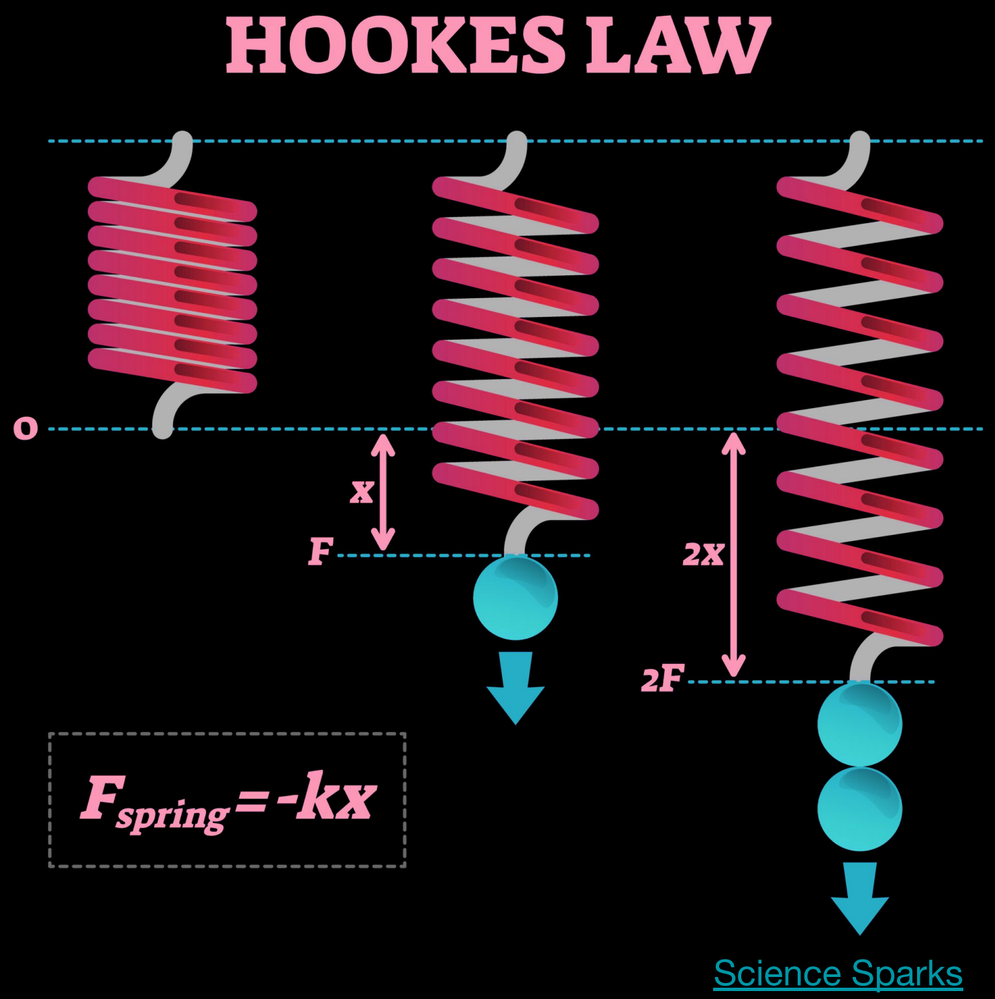
SoftRobotics Laboratory

Different types of softness



Guimarães, C.F., Gasperini, L., Marques, A.P. et al. Nat Rev Mater 5 (2020)

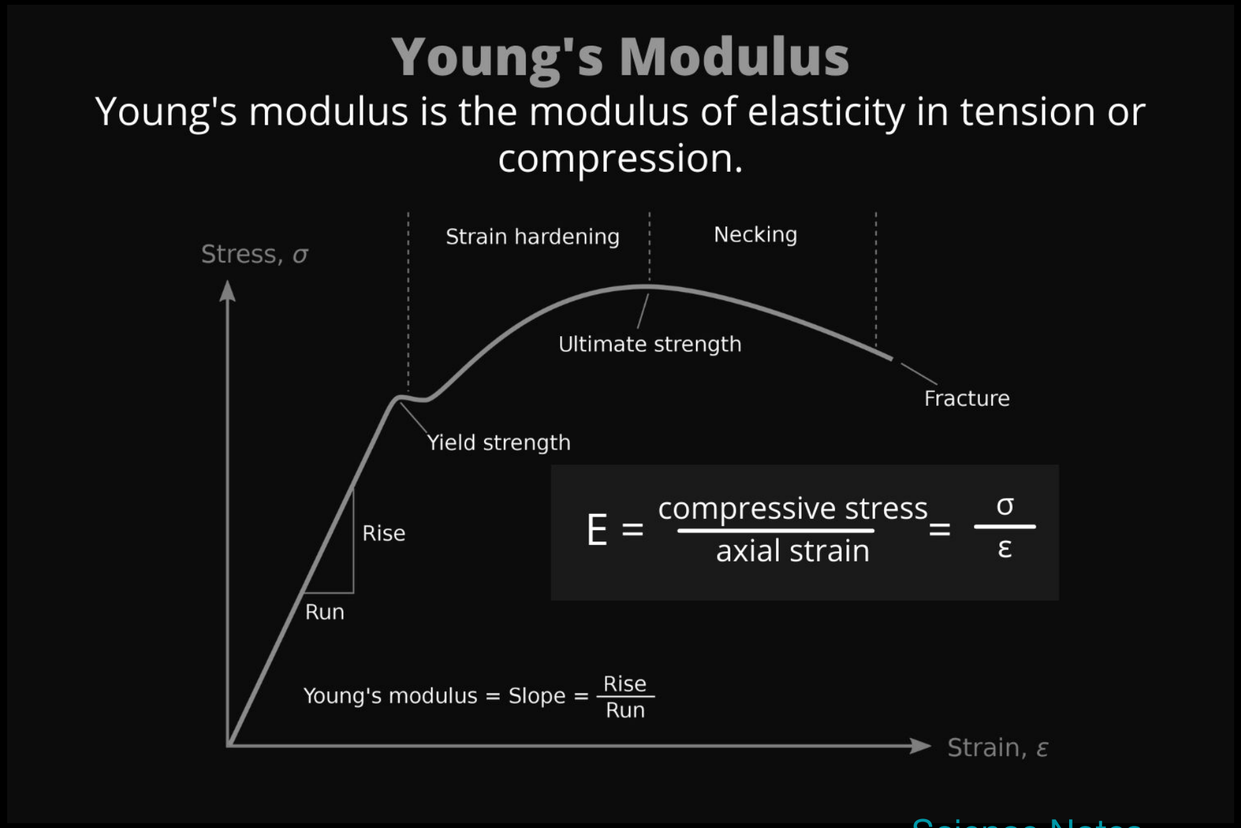
Hooke’s Law



In Hooke’s law, *k* represents the hardness of the spring. Similarly to springs we can try to find a numerical value that represents the behavior of an object when it is compressed or extended.

Bodies with a similar way of acting can be put into categories. Scientist then try to make a mathematical model that can simulate how the object acts.

Young’s Modulus

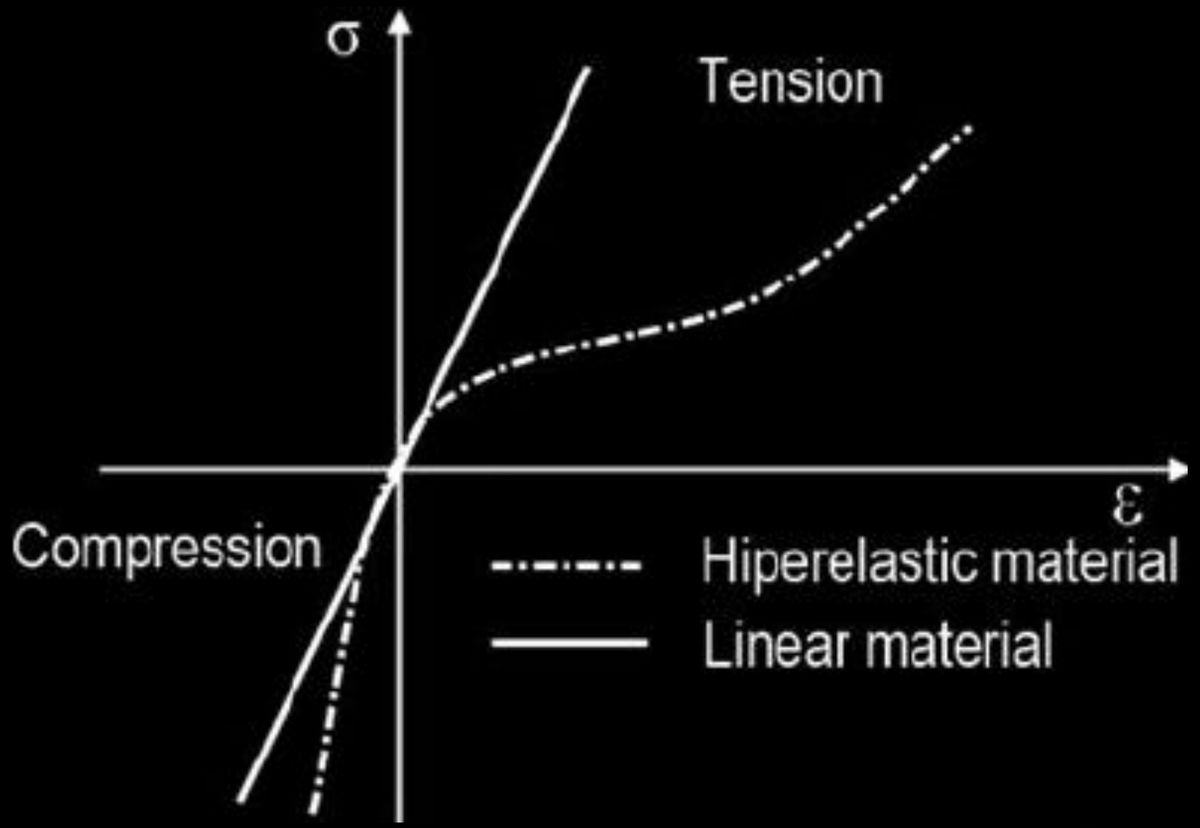


Science Notes

Similarly to Hooke’s Law, we express the modulus of elasticity in tension or compression of a body with the letter E.

For small strains, the object that follows Young’s Modulus stays in the linear part of the curve, this means that it can be extended and brought back to its original state without fracture.

Young’s Modulus



Quora

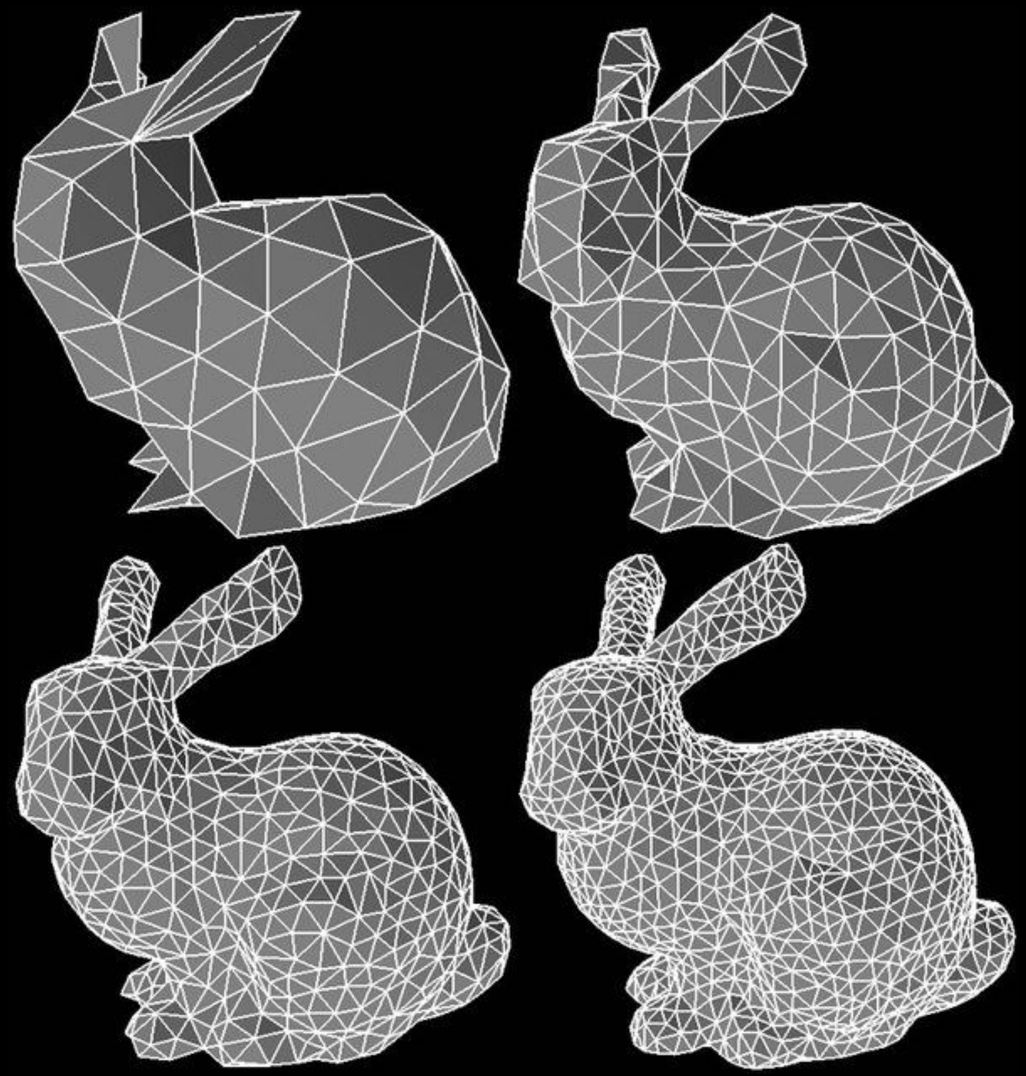
Similarly to Hooke’s Law, we express the modulus of elasticity in tension or compression of a body with the letter E.

For small strains, the object that follows Young’s Modulus stays in the linear part of the curve, this means that it can be extended and brought back to its original state without fracture.

Some materials can’t be approximated with precision with this model, therefore they need more advanced solutions.

Sofa models softness

Sofa: meshes



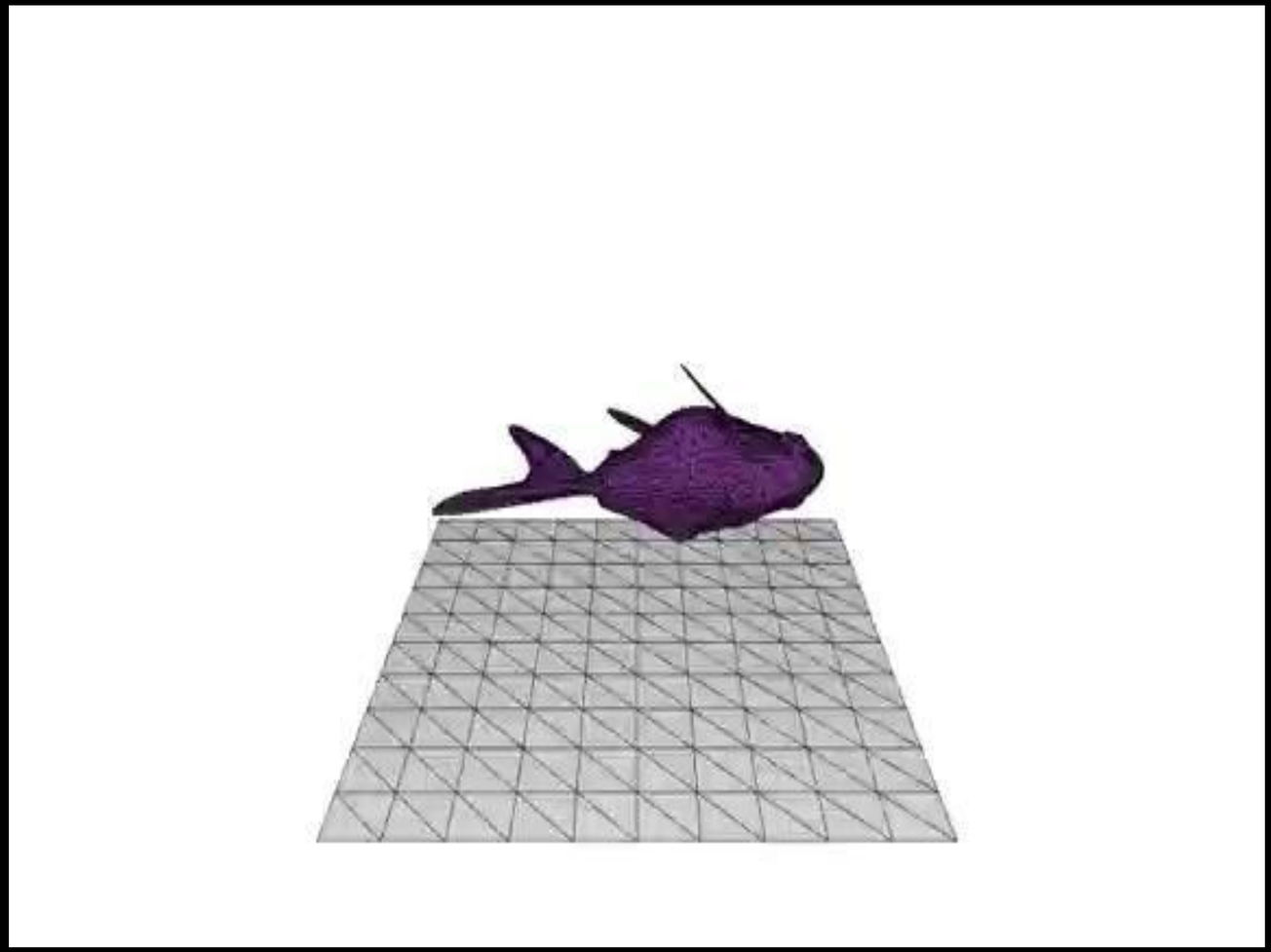
In 3D computer graphics and solid modeling, a polygon mesh is a collection of vertices, edges and faces that defines the shape of a polyhedral object. The faces usually consist of triangles (triangle mesh), quadrilaterals (quads), or other simple convex polygons (n-gons), since this simplifies rendering, but may also be more generally composed of concave polygons, or even polygons with holes.

ETH zürich SoftRobotics Laboratory

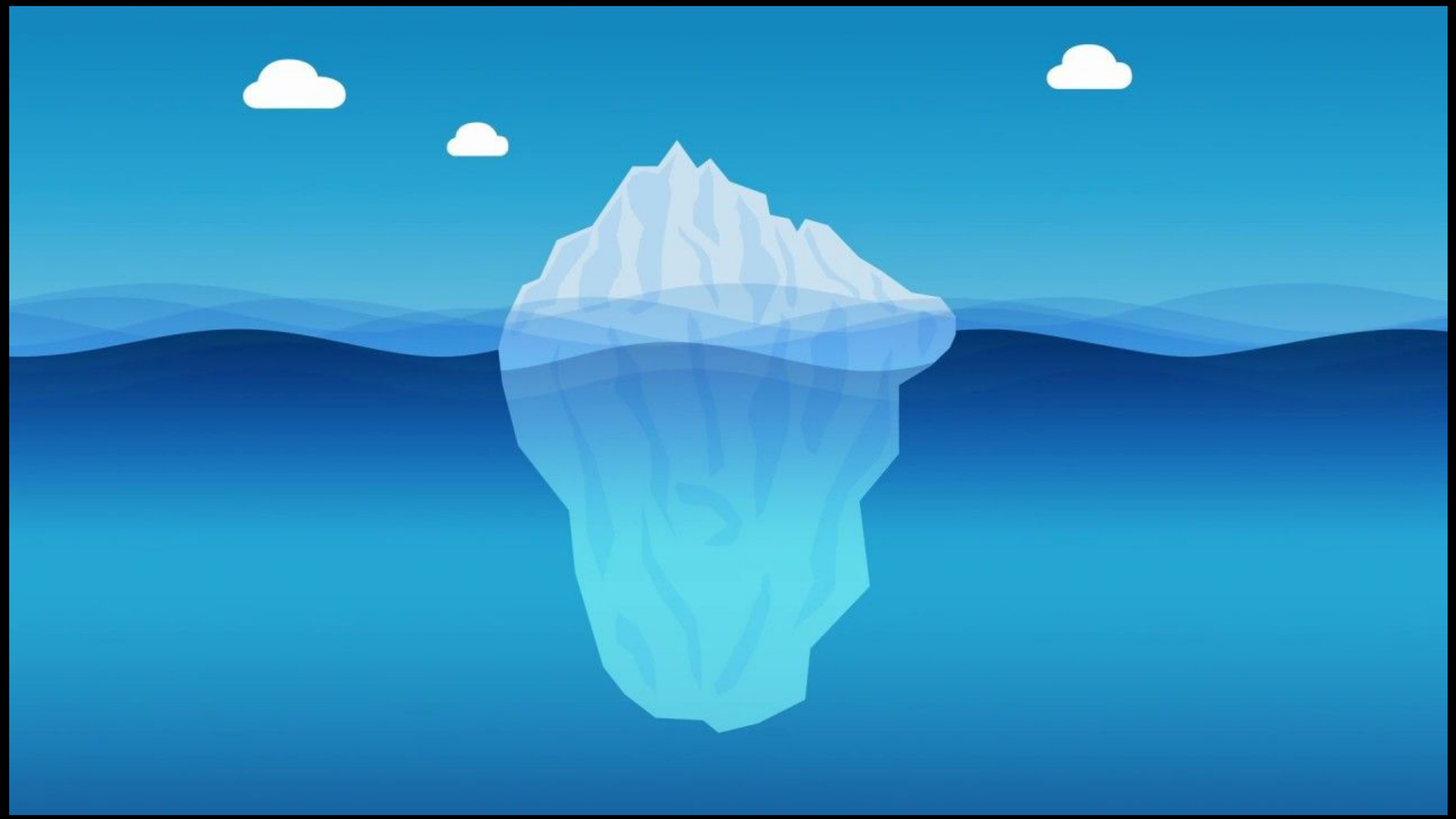
Quora

Source: Polygon mesh - Wikipedia

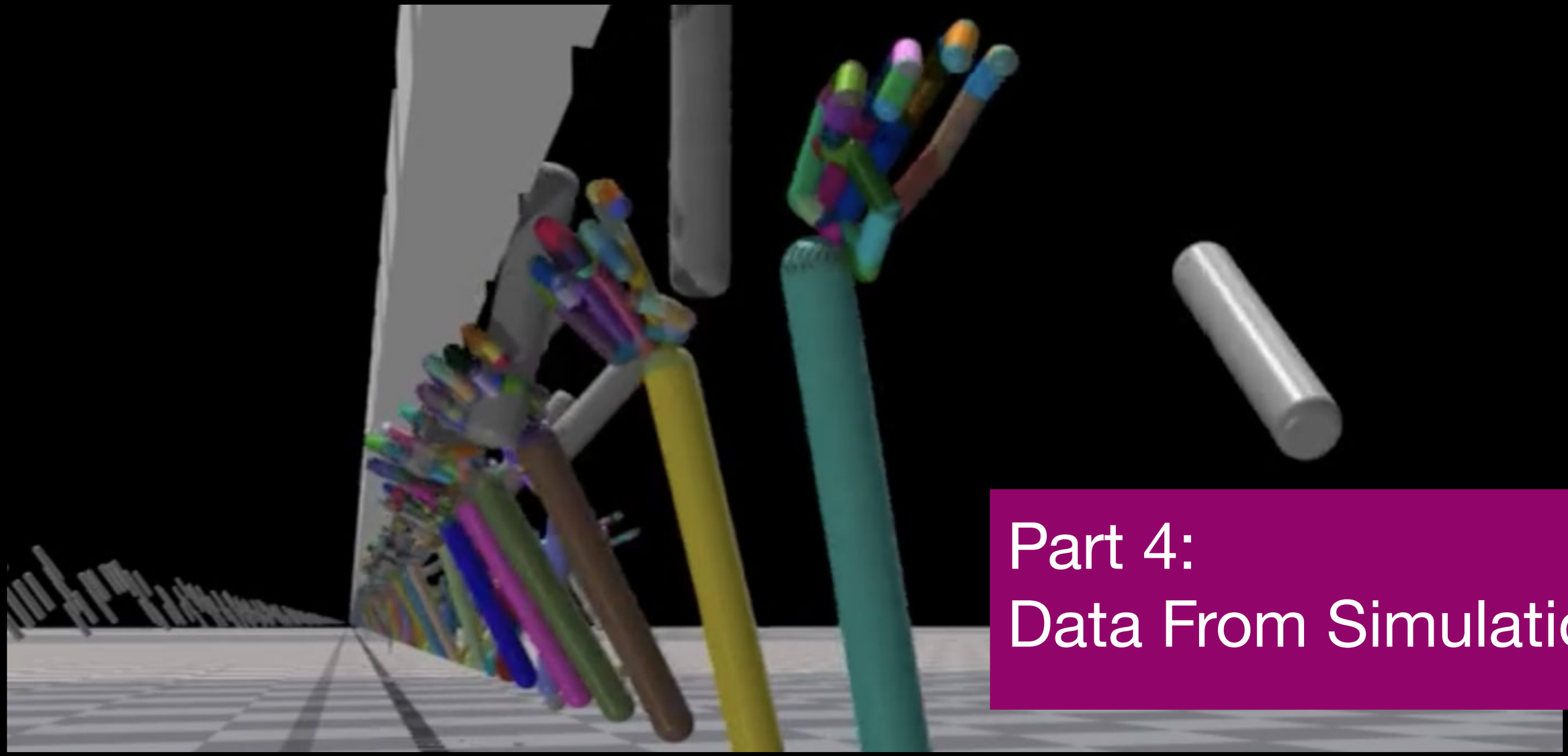
FEM



More can be said



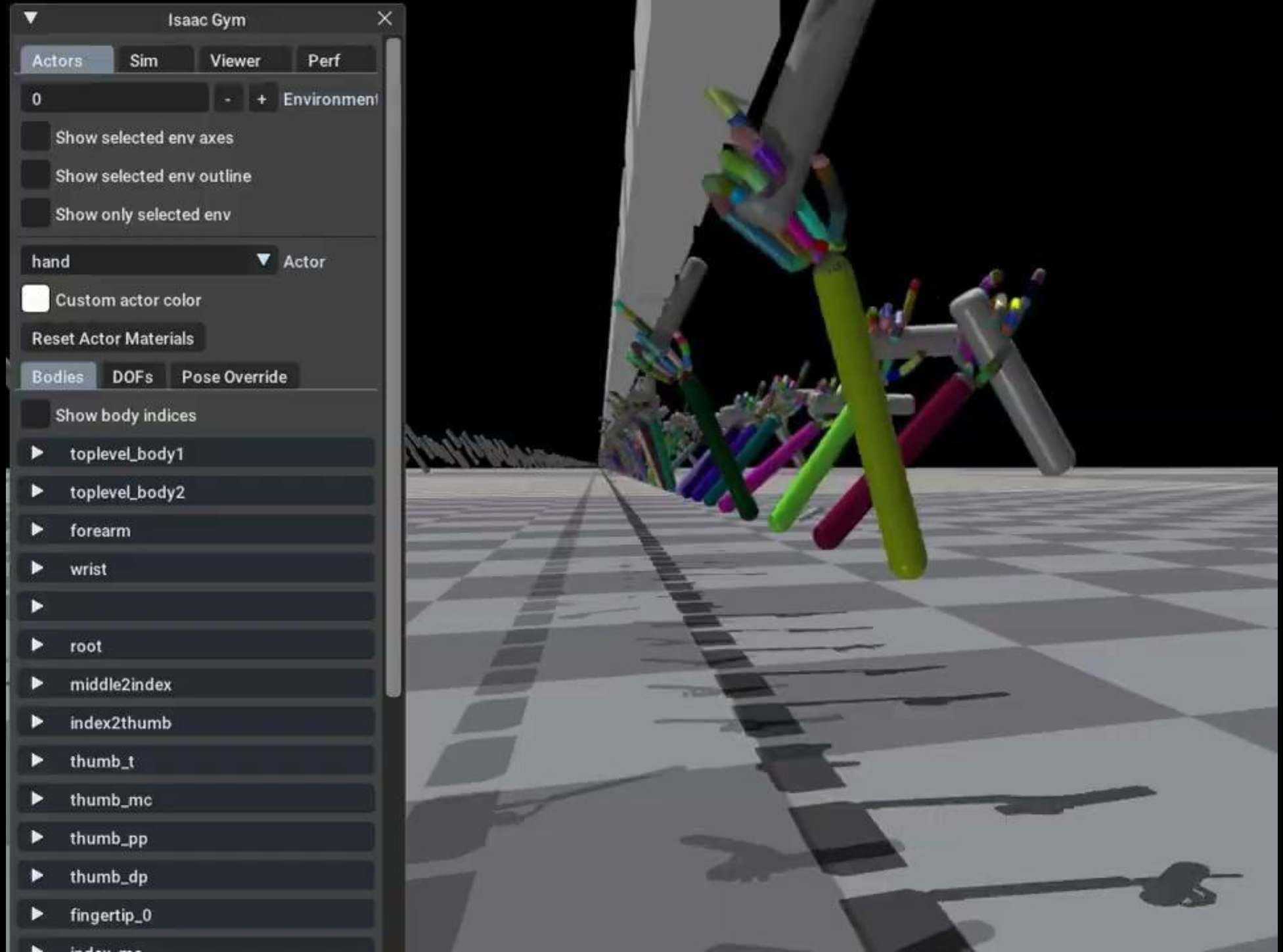
ETH zürich



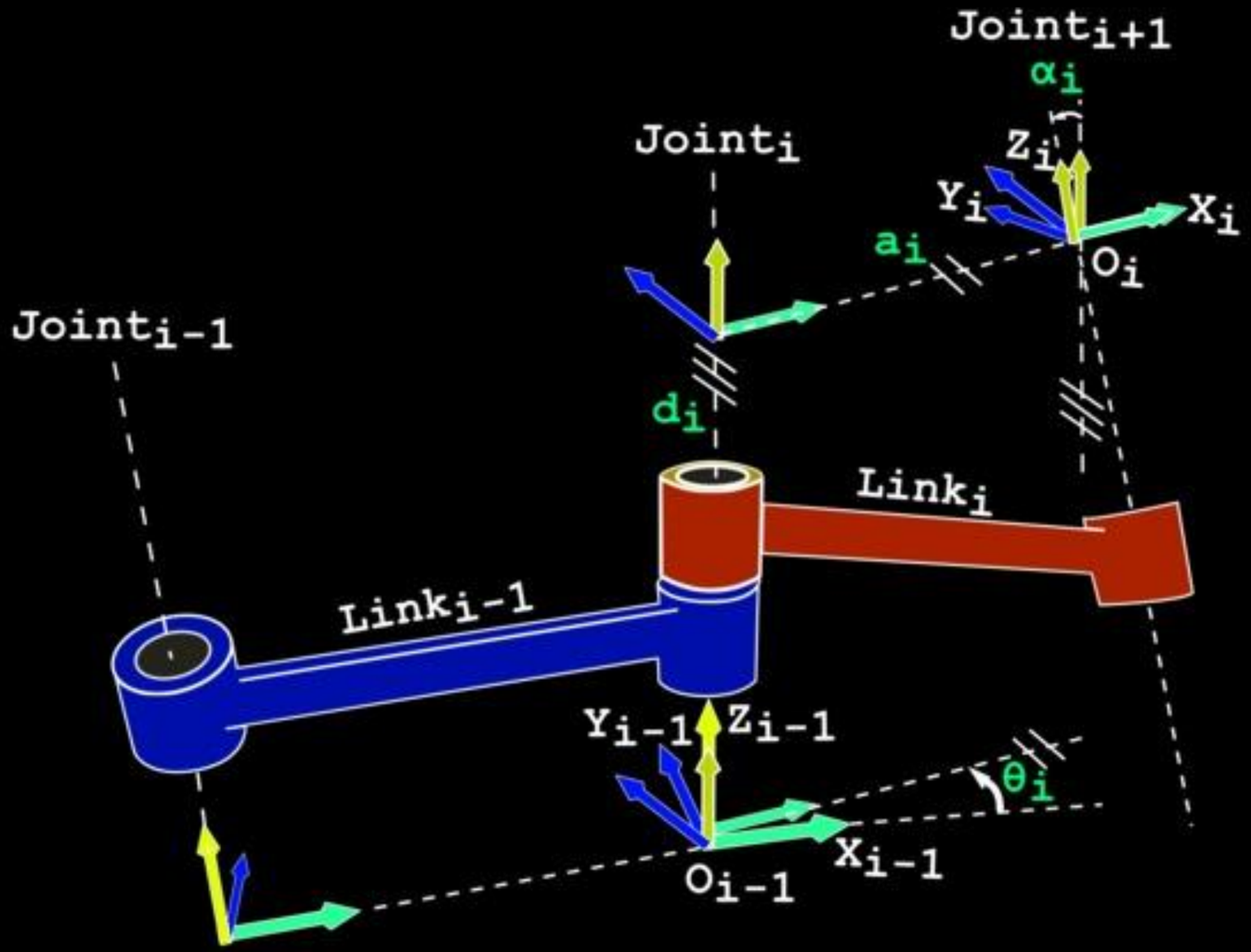
Part 4:
Data From Simulation

SoftRobotics Laboratory


Isaac Gym
Actors
Sim
Viewer
Perf
0
-
+
Environment
Show selected env axes
Show selected env outline
Show only selected env
hand
Actor
Custom actor color
Reset Actor Materials
Bodies
DOFs
Pose Override
Show body indices
virtual_root
root
palm_center
thumb_base
thumb_pp
thumb_mp
thumb_dp
thumb_fingertip
index_pp
index_mp


Isaac Gym
Actors
Sim
Viewer
Perf
0
-
+
Environment
Show selected env axes
Show selected env outline
Show only selected env
hand
Actor
Custom actor color
Reset Actor Materials
Bodies
DOFs
Pose Override
Show body indices
toplevel_body1
toplevel_body2
forearm
wrist
root
middle2index
index2thumb
thumb_t
thumb_mc
thumb_pp
thumb_dp
fingertip_0

What's next? Kinematics and Dynamics!



Wikimedia

Summary

- Simulation
  - Why do we use a simulator?
  - What is a simulator?
  - What simulators can we choose from? IsaacGym, Orbit, Drake, Gazebo
- Modelling of a hand
  - MJCF File
- Simulating softness with SOFA
  - Young’s modulus
  - FEM and meshes
- Data from simulation
  - What are the data that we can access? Positions, Forces, WandB

Useful Links

Create your own urdf file - ROS/Tutorials

xacro - ROS Wiki

How To Use Linux Screen

GitHub - NVIDIA-Omniverse/IsaacGymEnvs: Isaac Gym Reinforcement Learning Environments

XML Reference - MuJoCo Documentation

Modeling - MuJoCo Documentation

Actuator/Motor - MuJoCo Documentation

Spinning Up in Deep RL!

Isaac Gym - Preview Release | NVIDIA Developer

Omniverse Platform for OpenUSD Development and Collaboration | NVIDIA

Isaac ORBIT