

Lecture8: Mujoco Tutorial

Notes taken by [squarezhong](#)

Repo address: [squarezhong/SDM5008-Lecture-Notes](#)

Lecture8: Mujoco Tutorial

Introduction to Simulation

Definition

Three pillars of a simulator

Introduction to MuJoCo

Define a rigid body system

1. Where is it
2. How does it look
3. How does it connect to others
4. Its physical properties

Assets

Python Example

Introduction to Simulation

Definition

- Simulation: model physical world to do **prediction**
- All simulators essentially solve the **ODEs** and/or **PDEs** corresponding to a physical process of interest

Three pillars of a simulator

1. Constructing the differential equations/models
2. Solving differential equations
3. Visualization of the simulation results

Introduction to MuJoCo

Open-source robotics simulator maintained by Google Deepmind.

Define a rigid body system

All position is relative to the current frame.

1. Where is it

Body element (frame):

- `<pos>`: position
- `<euler>/<quat>/<axisangle>`: frame orientation relative to parent frame

2. How does it look

geom: sub-element of body

- `<type>`: sphere (default), plane, capsule, ellipsoid, cylinder, box, mesh, sdf
- `<fromto>`: from start position to end position
- `<rgba>`: color
- `<mass>`
- `<density>`: default "1000" (density of water in SI unit)

3. How does it connect to others

Joint: sub-element of body

a joint creates motion degrees of freedom between the body where it is defined and the body's parent.

- `<type>`:
 - `free`: 6 DoF
 - `slide`: linear
 - `hinge`: **rotation**
- `<pos>`
- `<axis>`: rotation axis for "hinge", direction of translation for "slide"

4. Its physical properties

- (default): infer from geometry attached to the body
- inertia sub-element
 - `<pos>`
 - `<orientation>`
 - `<mass>`
 - `<mass>`
 - `<diaginertia>`: real(3), diagonal entries of the inertia matrix
 - `<fullinertia>`: real(6), full inertia matrix

Assets

Assets are not in themselves model elements. Model elements can reference them

- `asset/mesh`: loaded from external files (e.g. STL ,OBJ .etc)
- `asset/material`: It can be referenced from skins, geoms, sites and tendons to set their appearance.

Python Example

MuJoCo accepts **MJCF** (Mujoco native) and **URDF** (Unified Robot Description Format) XML model files. Here we use URDF.

Refer to [Official XML Reference](#) is all you need.

```
1 <mujoco model="3R_robot">
2   <compiler angle="degree"/>
3   <asset>
4     <texture name="grid" type="2d" builtin="checker" rgb1=".1 .2 .3"
5       rgb2=".2 .3 .4" width="300" height="300" mark="none"/>
6     <material name="grid" texture="grid" texrepeat="6 6" texuniform="true"
reflectance=".2"/>
7   </asset>
8   <default>
9     <joint type="hinge" axis="0 0 1" limited="true"/>
10    <geom type="cylinder" size=".025 .1" />
11  </default>
12
13  <worldbody>
14    <light diffuse=".5 .5 .5" pos="0 0 3" dir="0 0 -1"/>
15    <geom type="plane" size="1 1 0.1" material="grid"/>
16    <body name="BaseLink" pos="0 0 0.1">
17      <geom type="cylinder" pos="0 0 0" size=".025 .1" />
18      <body name="link1" pos="0 0.1 0.125" euler="-90 0 0">
19        <joint name="joint1" pos="0 0 -0.1" range="-90 90" axis="0 1 0"/>
20        <geom pos="0 0 0" rgba=".6 .2 .2 1"/>
21        <site name="torque_site" pos="0 0.2 0"/>
22        <body name="link2" pos="0 0 0.2">
23          <joint name="joint2" pos="0 0 -0.1" range="-90 90" axis="0 1 0"/>
24          <geom rgba=".2 .6 1 1"/>
25          <site name="end_effector" pos="0 0 0.1" size="0.01"/>
26        </body>
27      </body>
28    </body>
29  </worldbody>
30 </mujoco>
```

Then use a Python script to load it:

```
1 import mujoco
2 import mujoco.viewer
3
4 model = mujoco.MjModel.from_xml_path("example.xml")
5 data = mujoco.MjData(model)
6
7 # Start the viewer to visualize the simulation
8 mujoco.viewer.launch(model, data)
```

It looks like:

File

Save xmlSave mjbPrint modelPrint dataQuitScreenshot

Option

HelpInfoProfilerSensorPause updateFullscreenVertical SyncBusy WaitSpacingTightColorDefaultFont200 %

Simulation

PauseRunResetReloadAlignCopy poseKey0Load keySave keyNoise scale0Noise rate0History0

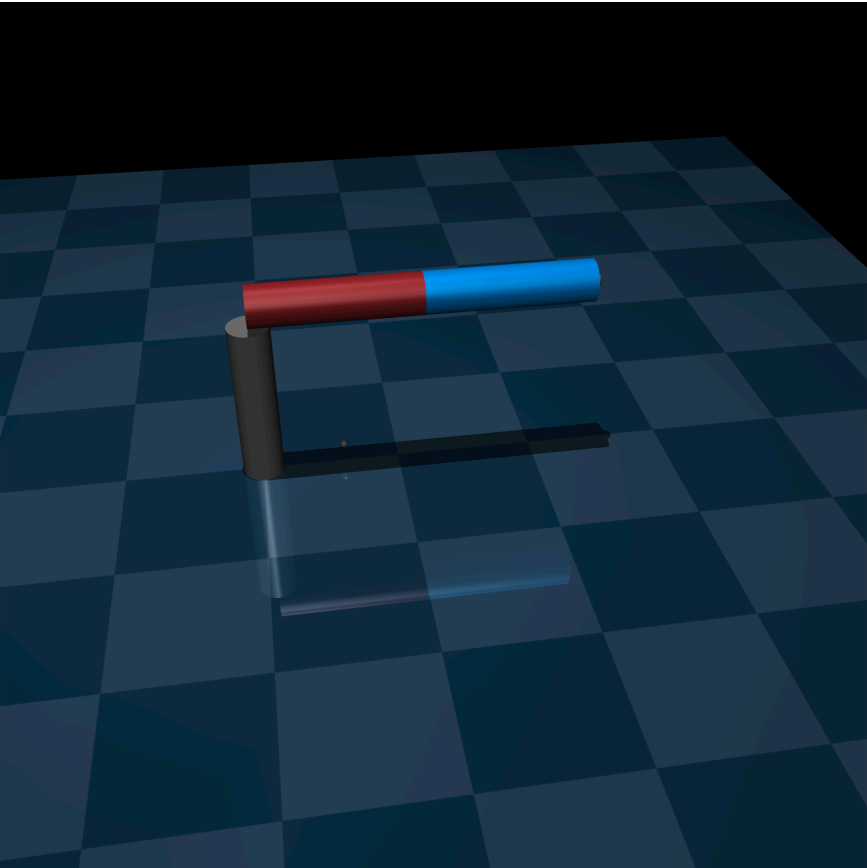
Watch

Physics

Rendering

Visualization

Group enable



Joint

Control