Computer Graphics

10 - Kinematics & Animation

Yoonsang Lee
Spring 2022

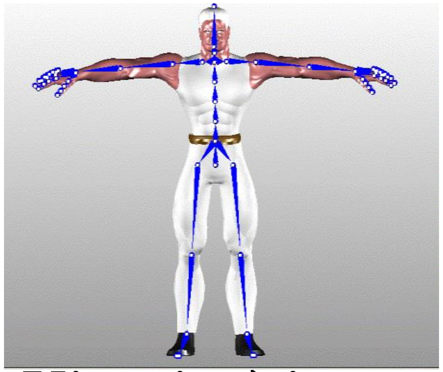
Topics Covered

- Forward Kinematics
- Introduction to Character Animation
- Motion Capture Data

Forward Kinematics

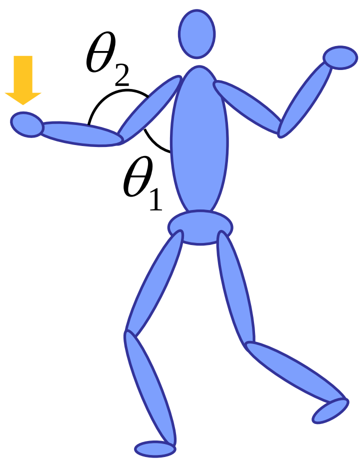
Kinematics

- *Kinematics* is
  - Study of **motion** of objects (or groups of objects), **without considering mass or forces**
  - In computer graphics, it's about how to move skeletons
    - Forward kinematics
    - Inverse kinematics



- By contrast, *Dynamics (or Kinetics)* is
  - Study of the **relationship between motion and its causes, specifically, forces and mass**

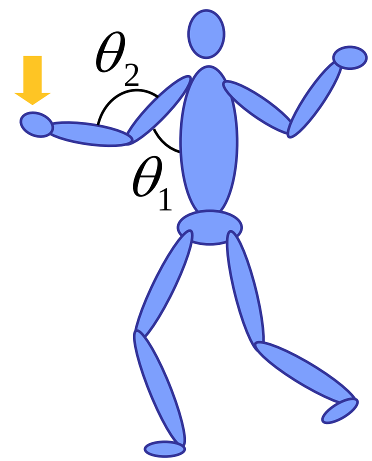
Kinematics



$$(\mathbf{p}, \mathbf{q}) = \mathrm{F}(\theta_i)$$

Forward Kinematics

: Given joint angles,
compute the position &
orientation of end-effector



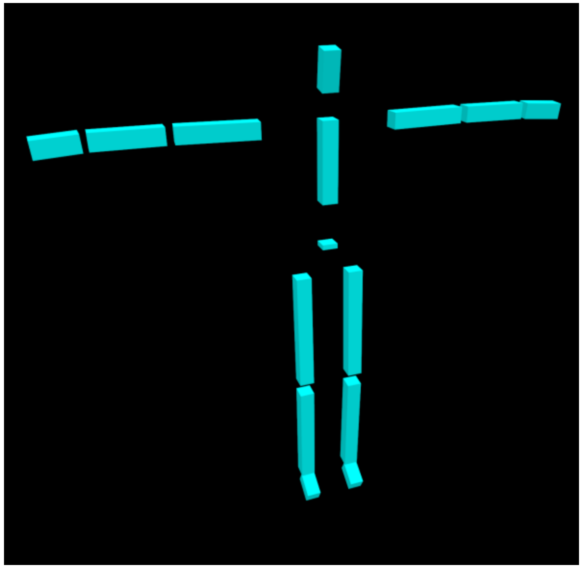
$$\theta_i = \mathrm{F}^{-1}(\mathbf{p}, \mathbf{q})$$

Inverse Kinematics

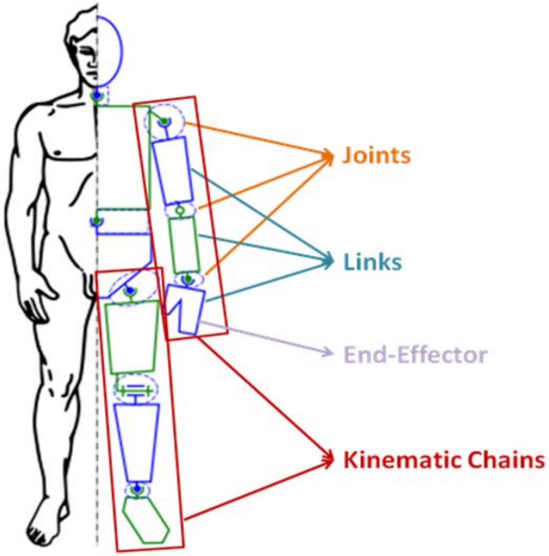
: Given the position &
orientation of end-effector,
compute joint angles

Articulated Body

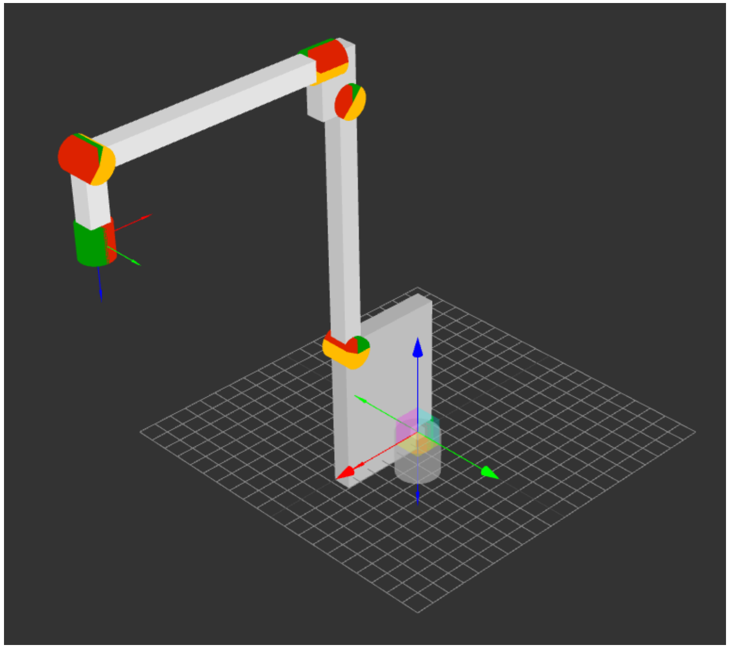
- A common type of hierarchical model used in CG is an *articulated body*
  - that has objects that are connected end to end to form multibody jointed chains.
  - a.k.a. *kinematic chain*, *linkage* (robotics)



- Terminologies
  - ***Joint*** - a connection between two objects which allows some motion
  - ***Link*** - a rigid object between joints
  - ***End effector*** - a free end of a kinematic chain



[Practice] FK / IK Online Demo

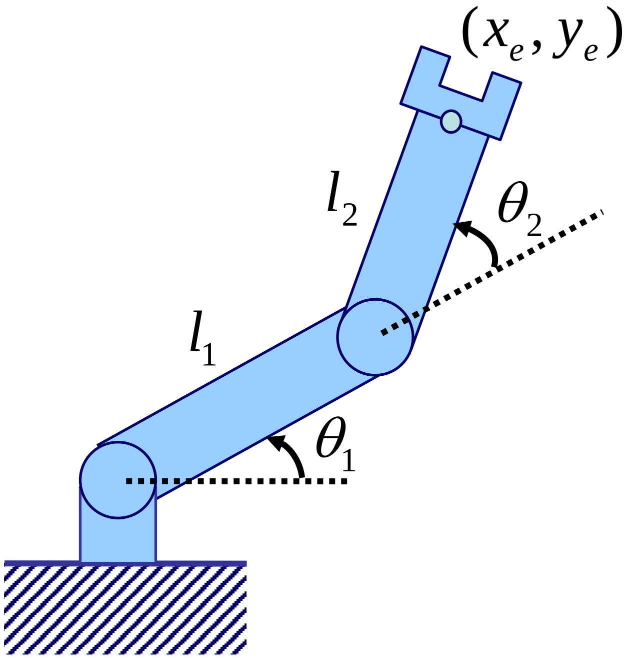


http://robot.glumb.de/

- Forward kinematics : Open “angles” menu and change values
- Inverse kinematics : Move the end-effector position by mouse dragging

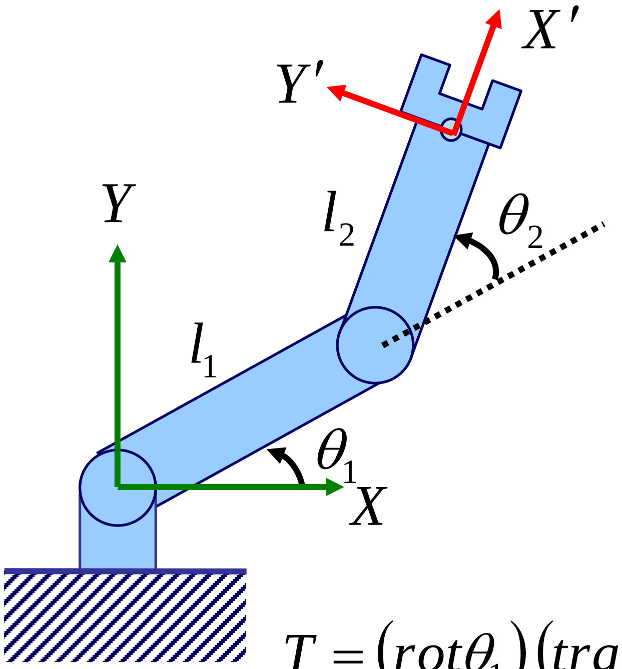
Forward Kinematics: A Simple Example

- A simple robot arm in 2-dimensional space
  - 2 revolute joints
  - Joint angles are known
  - Compute the position of the end-effector



$$x_e = l_1 \cos\theta_1 + l_2 \cos(\theta_1 + \theta_2)$$
$$y_e = l_1 \sin\theta_1 + l_2 \sin(\theta_1 + \theta_2)$$

A Chain of Transformations



$$\begin{pmatrix} x_e \\ y_e \\ 1 \end{pmatrix} = \begin{pmatrix} & & \\ & T & \\ & & \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$$

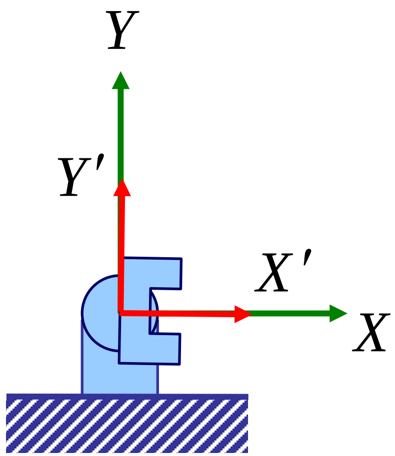
$$T = (rot\theta_1)(transl_1)(rot\theta_2)(transl_2)$$

$$= \begin{pmatrix} \cos\theta_1 & -\sin\theta_1 & 0 \\ \sin\theta_1 & \cos\theta_1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & l_1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \cos\theta_2 & -\sin\theta_2 & 0 \\ \sin\theta_2 & \cos\theta_2 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & l_2 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Thinking of Transformations

- In a view of body-attached coordinate system

(=local coordinate system of the end-effector body)

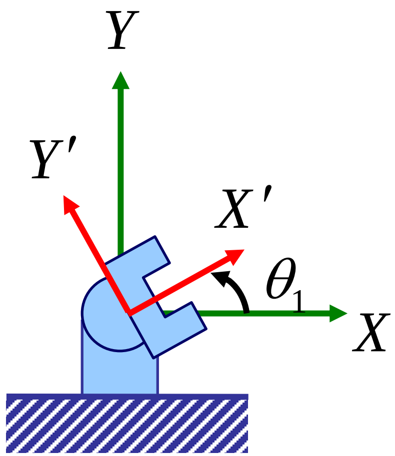


$$T = (rot\theta_1)(transl_1)(rot\theta_2)(transl_2)$$

$$= \begin{pmatrix} \cos\theta_1 & -\sin\theta_1 & 0 \\ \sin\theta_1 & \cos\theta_1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & l_1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \cos\theta_2 & -\sin\theta_2 & 0 \\ \sin\theta_2 & \cos\theta_2 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & l_2 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Thinking of Transformations

- In a view of body-attached coordinate system

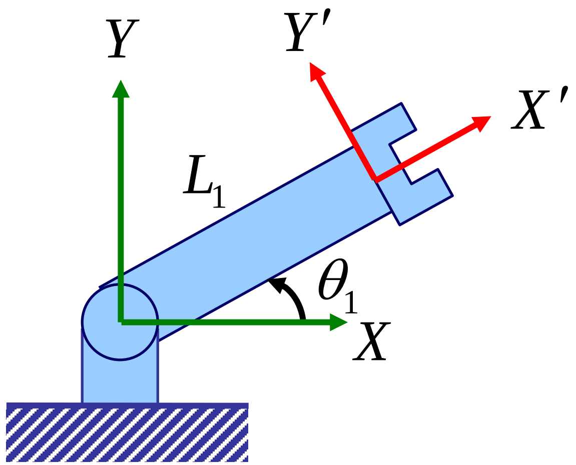


$$T = (rot\theta_1)(transl_1)(rot\theta_2)(transl_2)$$

$$= \begin{pmatrix} \cos\theta_1 & -\sin\theta_1 & 0 \\ \sin\theta_1 & \cos\theta_1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & l_1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \cos\theta_2 & -\sin\theta_2 & 0 \\ \sin\theta_2 & \cos\theta_2 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & l_2 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Thinking of Transformations

- In a view of body-attached coordinate system

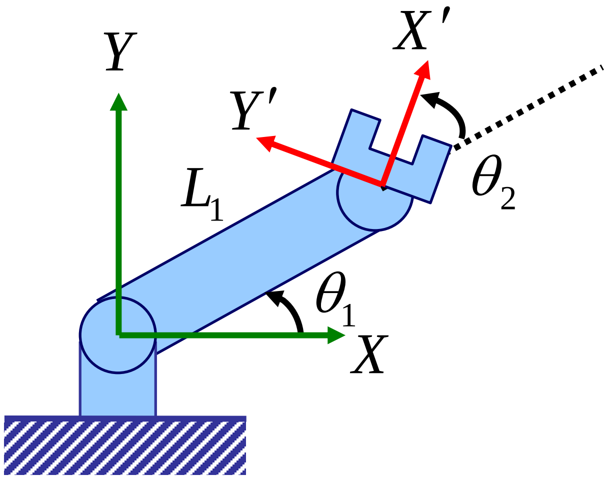


$$T = (rot\theta_1)(transl_1)(rot\theta_2)(transl_2)$$

$$= \begin{pmatrix} \cos\theta_1 & -\sin\theta_1 & 0 \\ \sin\theta_1 & \cos\theta_1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & l_1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \cos\theta_2 & -\sin\theta_2 & 0 \\ \sin\theta_2 & \cos\theta_2 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & l_2 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Thinking of Transformations

- In a view of body-attached coordinate system

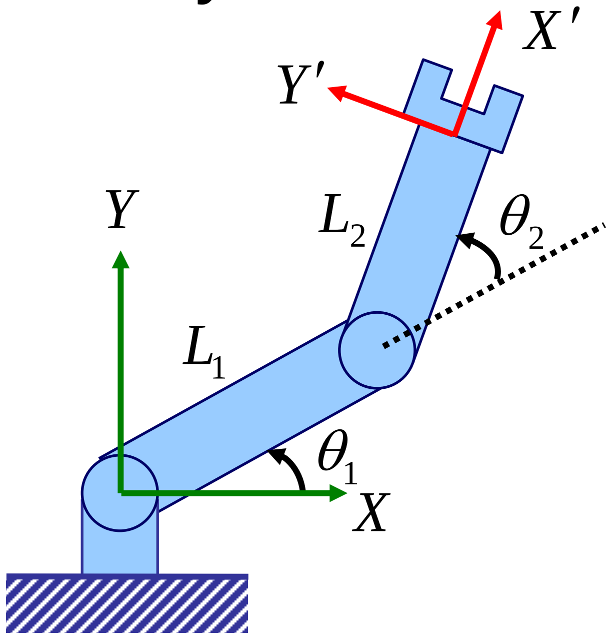


$$T = (rot\theta_1)(transl_1)(rot\theta_2)(transl_2)$$

$$= \begin{pmatrix} \cos\theta_1 & -\sin\theta_1 & 0 \\ \sin\theta_1 & \cos\theta_1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & l_1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \cos\theta_2 & -\sin\theta_2 & 0 \\ \sin\theta_2 & \cos\theta_2 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & l_2 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Thinking of Transformations

- In a view of body-attached coordinate system

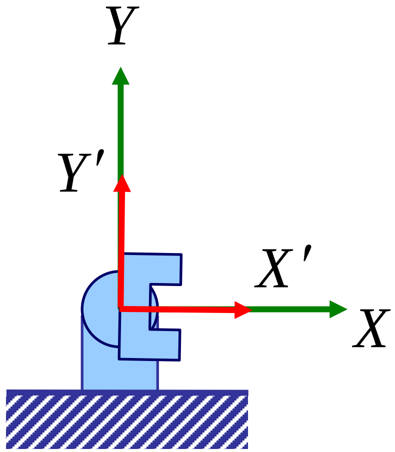


$$T = (rot\theta_1)(transl_1)(rot\theta_2)(transl_2)$$

$$= \begin{pmatrix} \cos\theta_1 & -\sin\theta_1 & 0 \\ \sin\theta_1 & \cos\theta_1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & l_1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \cos\theta_2 & -\sin\theta_2 & 0 \\ \sin\theta_2 & \cos\theta_2 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & l_2 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Thinking of Transformations

- In a view of global coordinate system

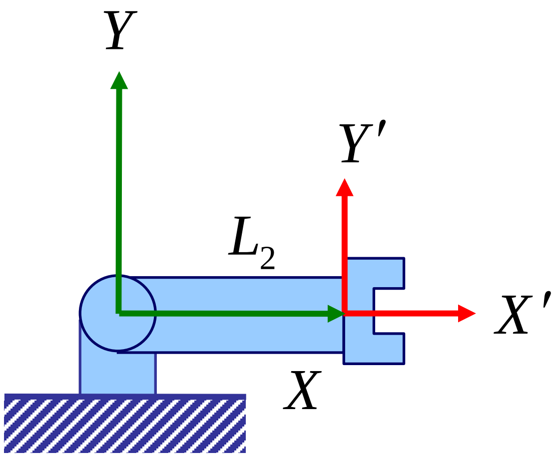


$$T = (rot\theta_1)(transl_1)(rot\theta_2)(transl_2)$$

$$= \begin{pmatrix} \cos\theta_1 & -\sin\theta_1 & 0 \\ \sin\theta_1 & \cos\theta_1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & l_1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \cos\theta_2 & -\sin\theta_2 & 0 \\ \sin\theta_2 & \cos\theta_2 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & l_2 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Thinking of Transformations

- In a view of global coordinate system

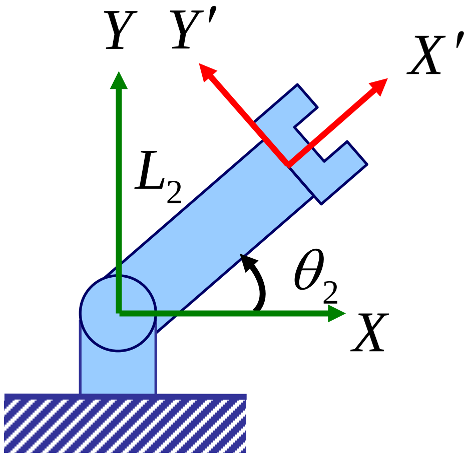


$$T = (rot\theta_1)(transl_1)(rot\theta_2)(transl_2)$$

$$= \begin{pmatrix} \cos\theta_1 & -\sin\theta_1 & 0 \\ \sin\theta_1 & \cos\theta_1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & l_1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \cos\theta_2 & -\sin\theta_2 & 0 \\ \sin\theta_2 & \cos\theta_2 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & l_2 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Thinking of Transformations

- In a view of global coordinate system

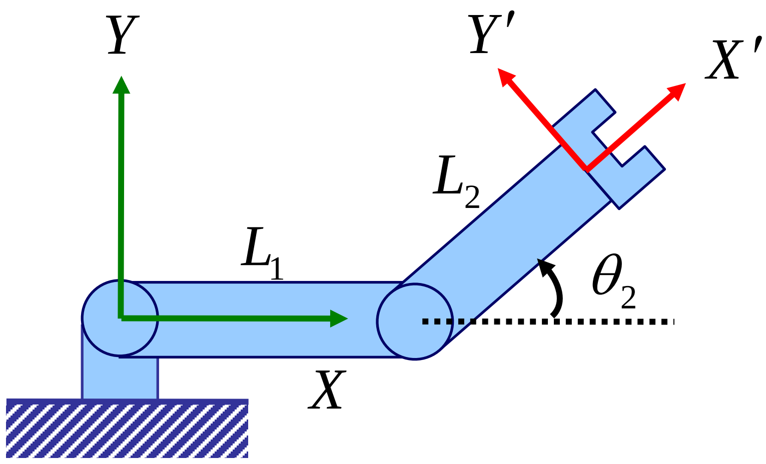


$$T = (rot\theta_1)(transl_1)(rot\theta_2)(transl_2)$$

$$= \begin{pmatrix} \cos\theta_1 & -\sin\theta_1 & 0 \\ \sin\theta_1 & \cos\theta_1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & l_1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \cos\theta_2 & -\sin\theta_2 & 0 \\ \sin\theta_2 & \cos\theta_2 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & l_2 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Thinking of Transformations

- In a view of global coordinate system

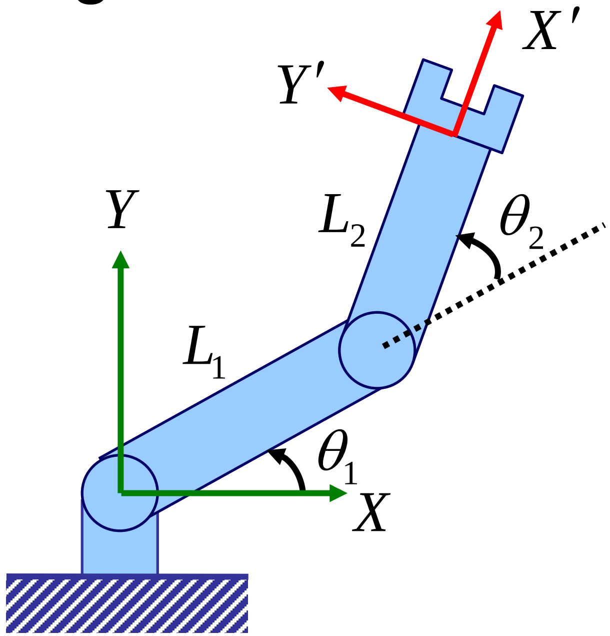


$$T = (rot\theta_1)(transl_1)(rot\theta_2)(transl_2)$$

$$= \begin{pmatrix} \cos\theta_1 & -\sin\theta_1 & 0 \\ \sin\theta_1 & \cos\theta_1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & l_1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \cos\theta_2 & -\sin\theta_2 & 0 \\ \sin\theta_2 & \cos\theta_2 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & l_2 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Thinking of Transformations

- In a view of global coordinate system



$$T = \left(rot\theta_1\right)\left(transl_1\right)\left(rot\theta_2\right)\left(transl_2\right)$$

$$= \begin{pmatrix} \cos\theta_1 & -\sin\theta_1 & 0 \\ \sin\theta_1 & \cos\theta_1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & l_1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \cos\theta_2 & -\sin\theta_2 & 0 \\ \sin\theta_2 & \cos\theta_2 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & l_2 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Quiz #1

- Go to https://www.slido.com/
- Join #**cg-ys**
- Click “Polls”

- Submit your answer in the following format:
  - **Student ID: Your answer**
  - **e.g. 2017123456: 4)**

- Note that you must submit all quiz answers in the above format to be checked for “attendance”.

Forward Kinematics Map

- A *forward kinematics map* is a mapping from **joint angles** to **end effector position & orientation.**

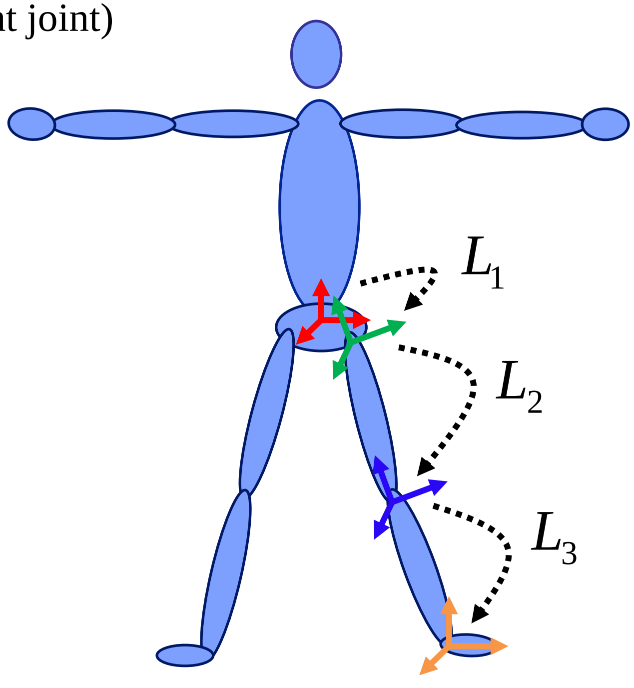
- Usually, a *forward kinematics map* T is an alternating multiple of ...
- **Joint transformations** that represents joint movement (**time-varying**)
  - Usually rotations
- **Link transformations** that defines a frame relative to its parent (**static**)
  - Usually translations (joint offset from its parent joint)

translation to 1st joint

$$T = J_0 L_1 J_1 L_2 J_2 L_3 J_3$$

rotation of 1st joint

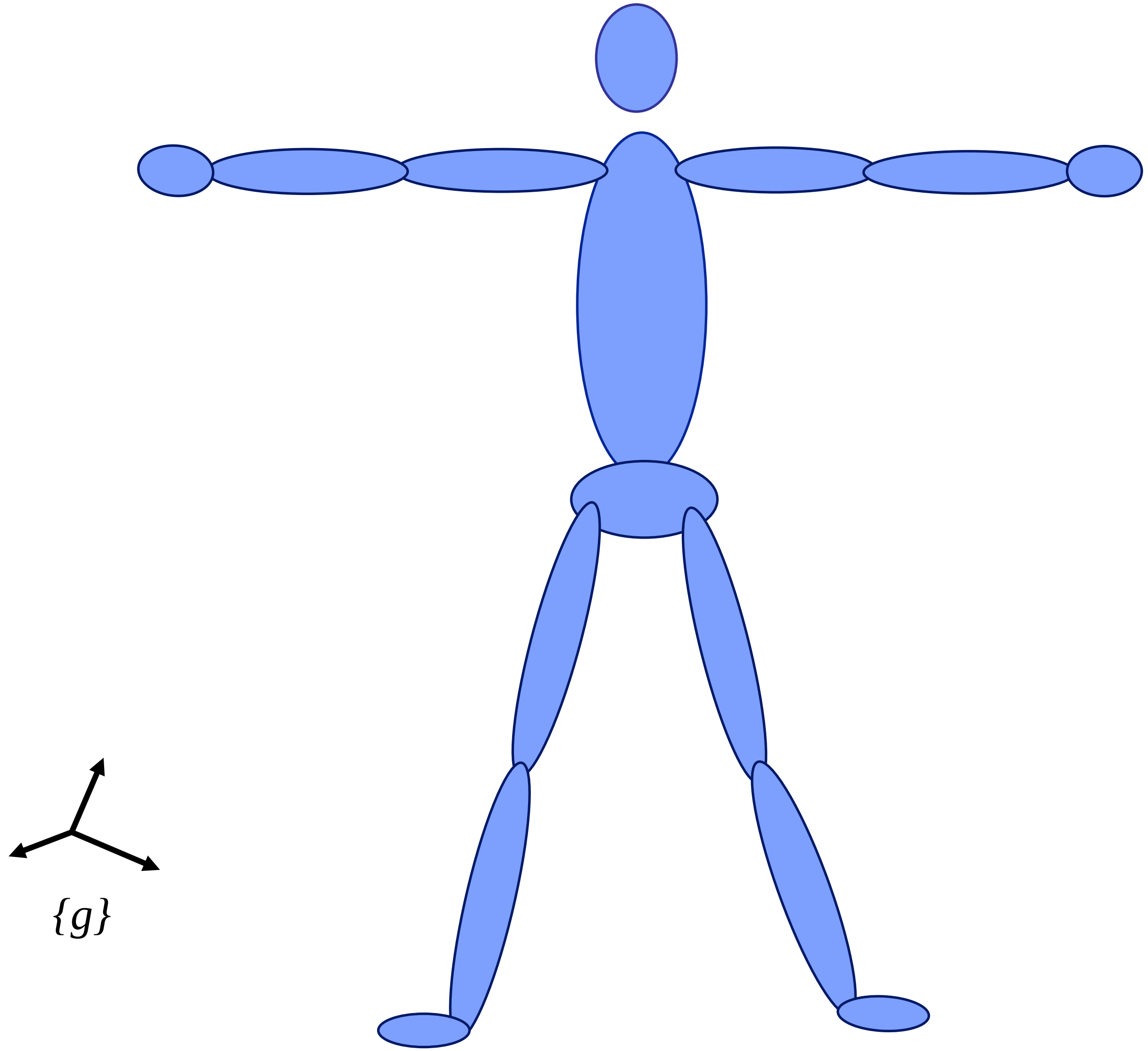
position & orientation of 0th joint (the root segment)



Forward Kinematics Map

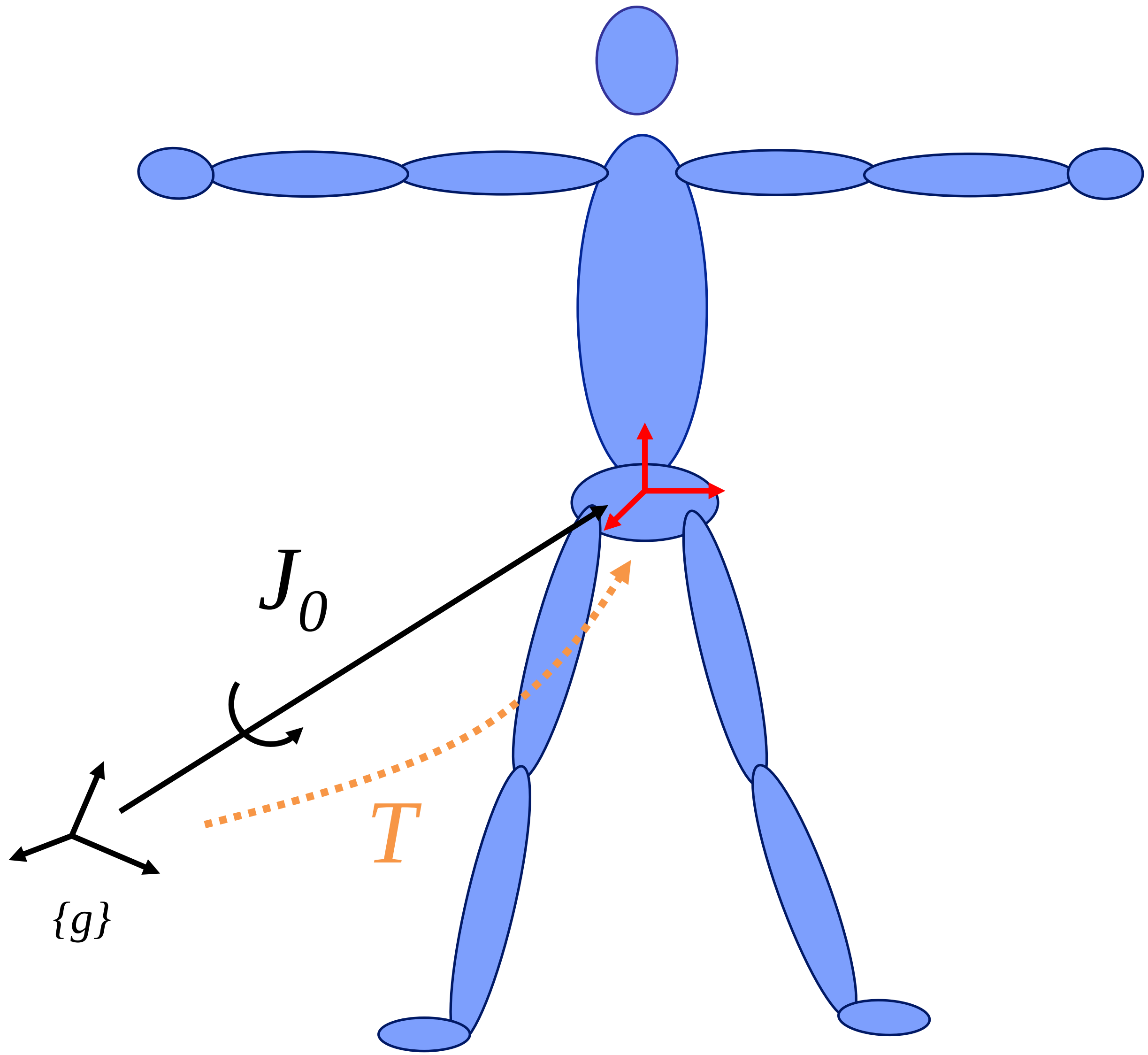
current frame

$$T = \mathrm{I}$$



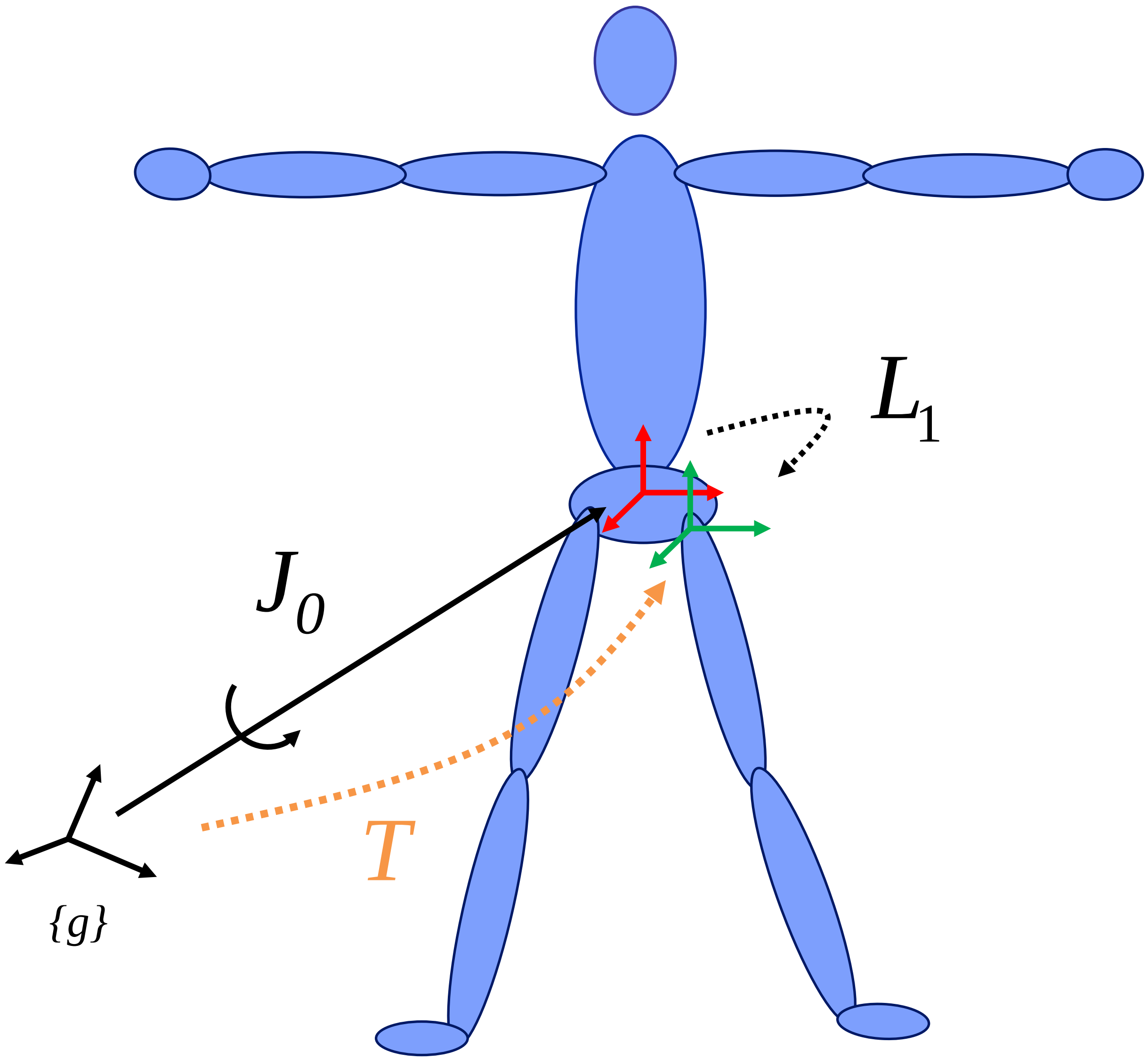
Forward Kinematics Map

$$T = J_0$$



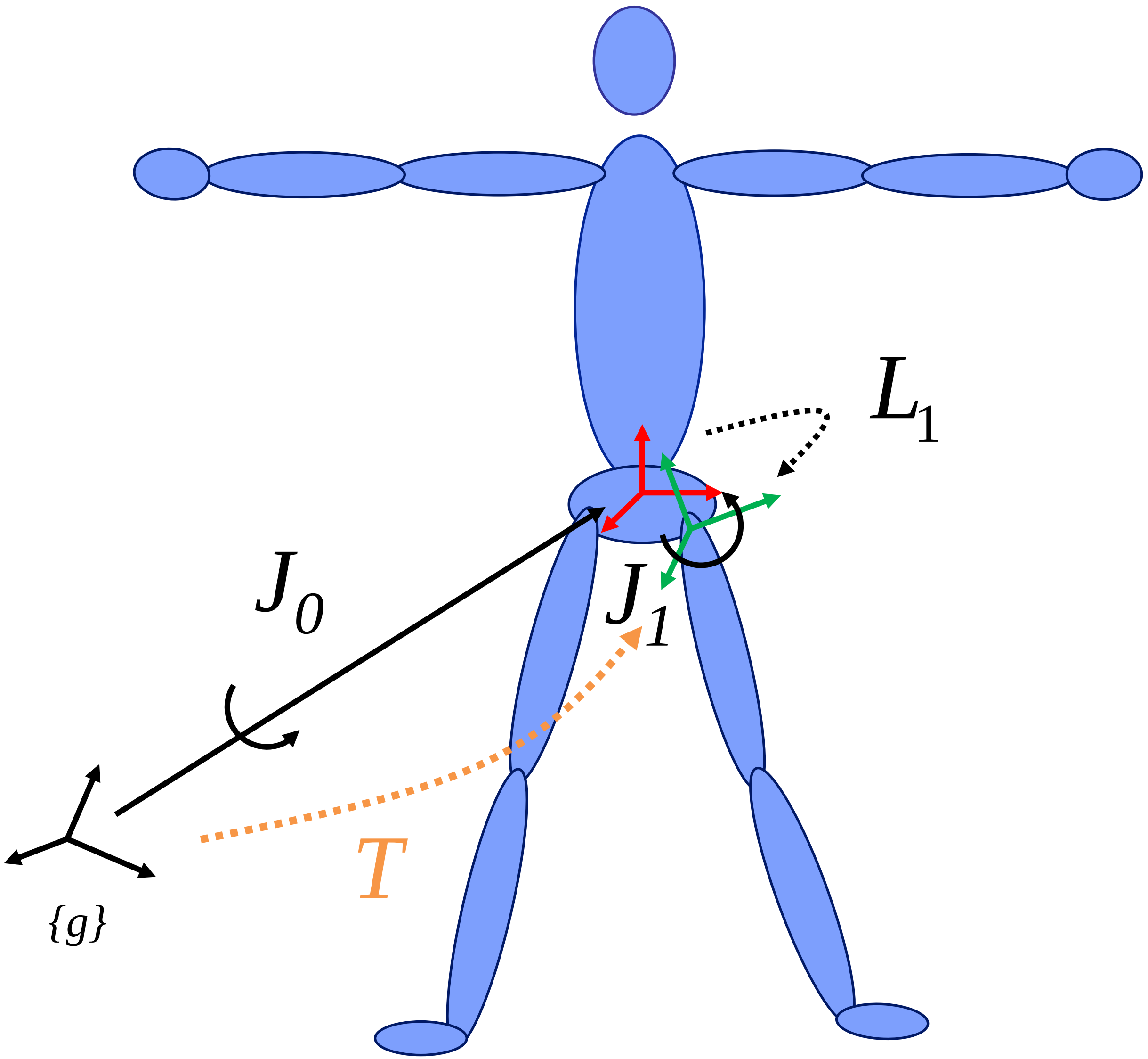
Forward Kinematics Map

$$T = J_0 L_1$$



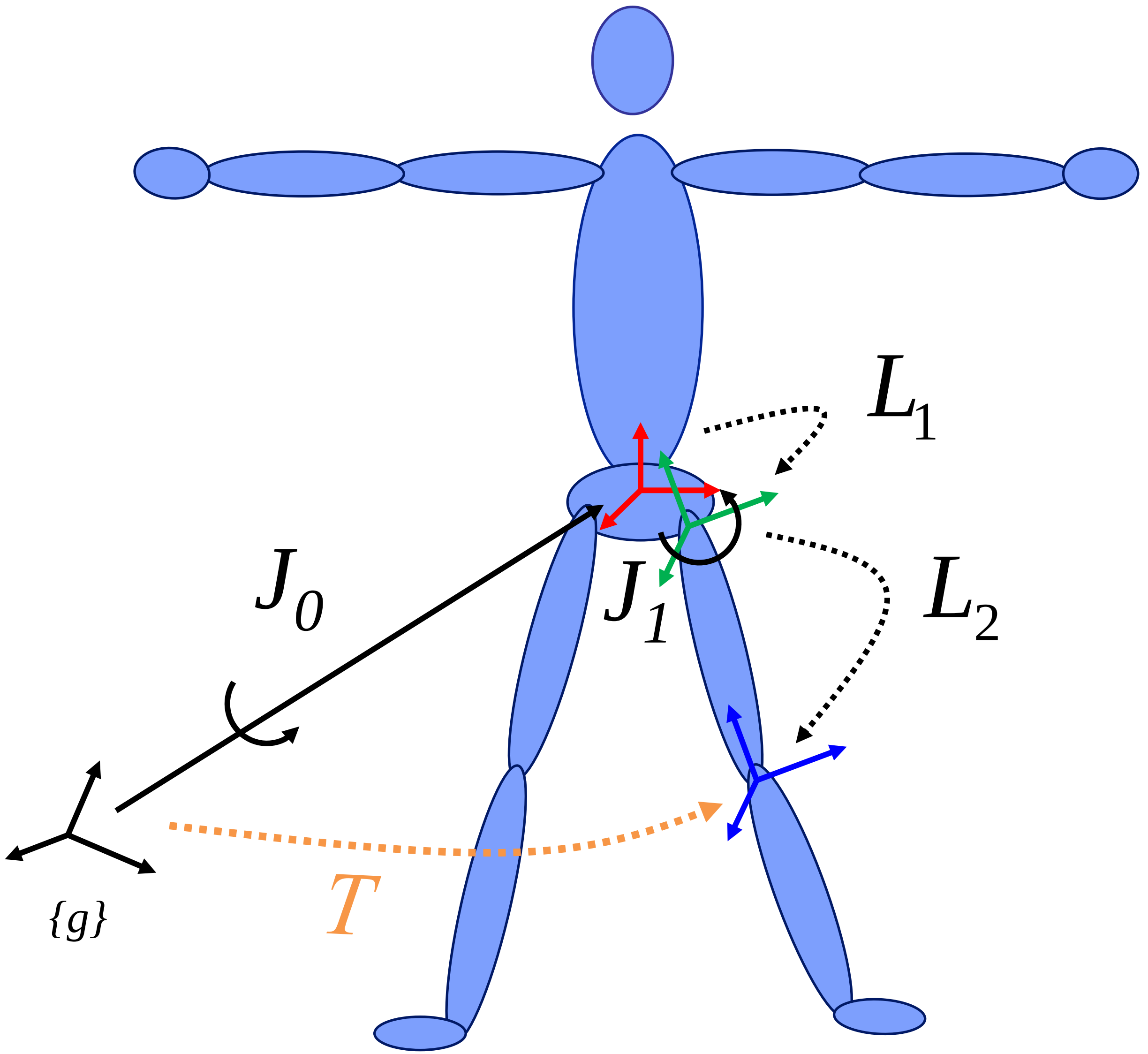
Forward Kinematics Map

$$T = J_0 L_1 J_1$$



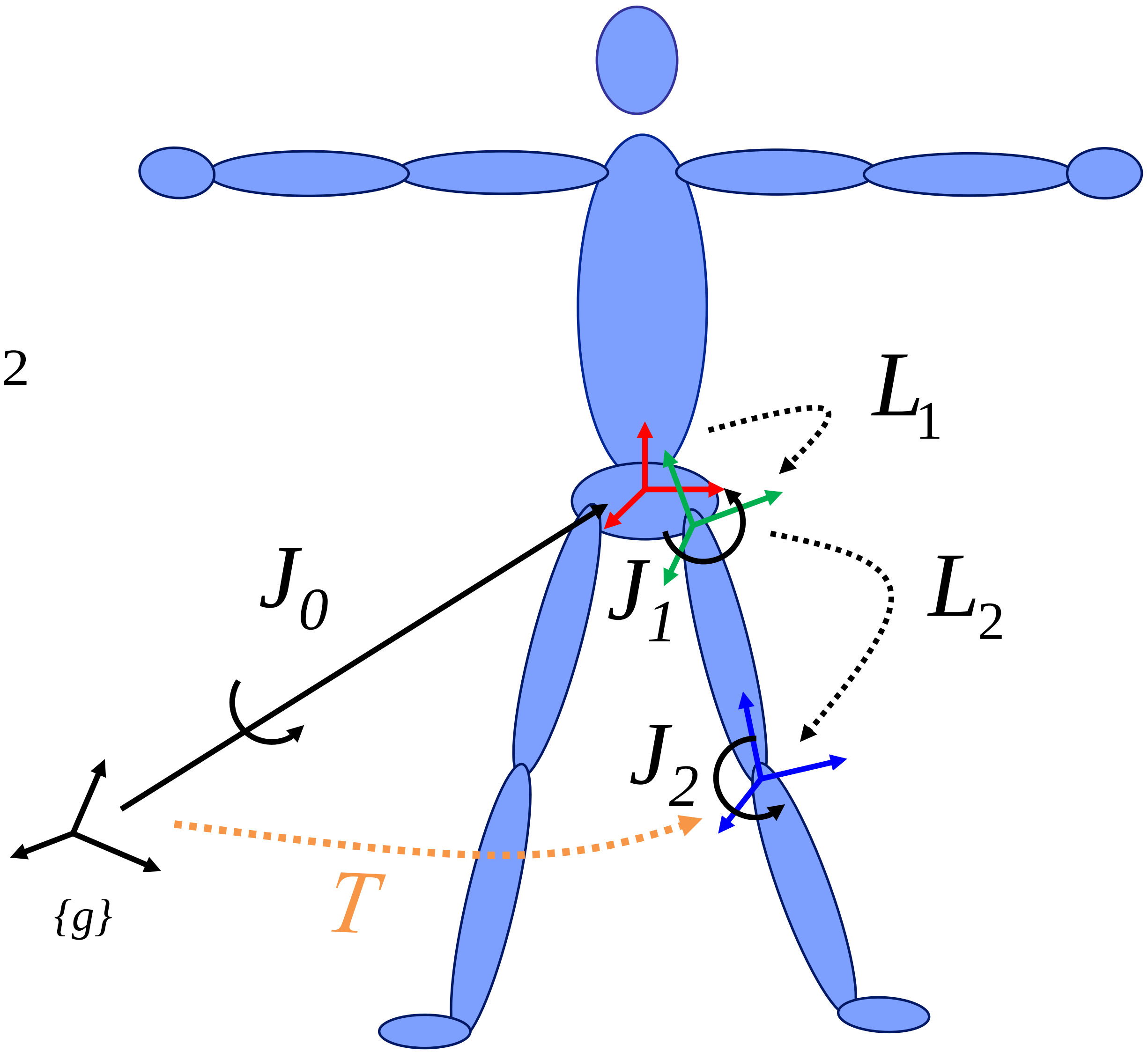
Forward Kinematics Map

$$T = J_0 L_1 J_1 L_2$$



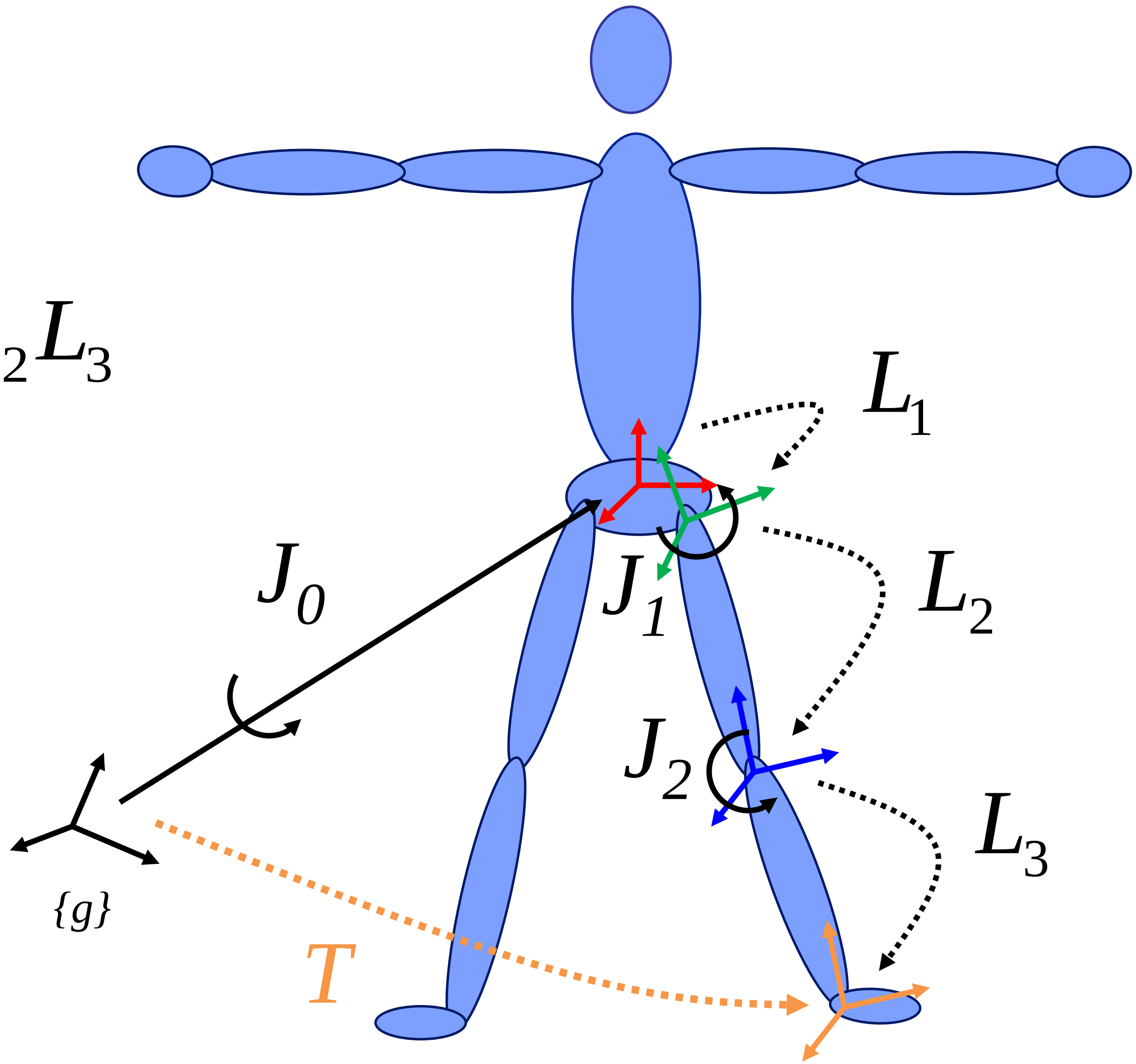
Forward Kinematics Map

$$T = J_0 L_1 J_1 L_2 J_2$$



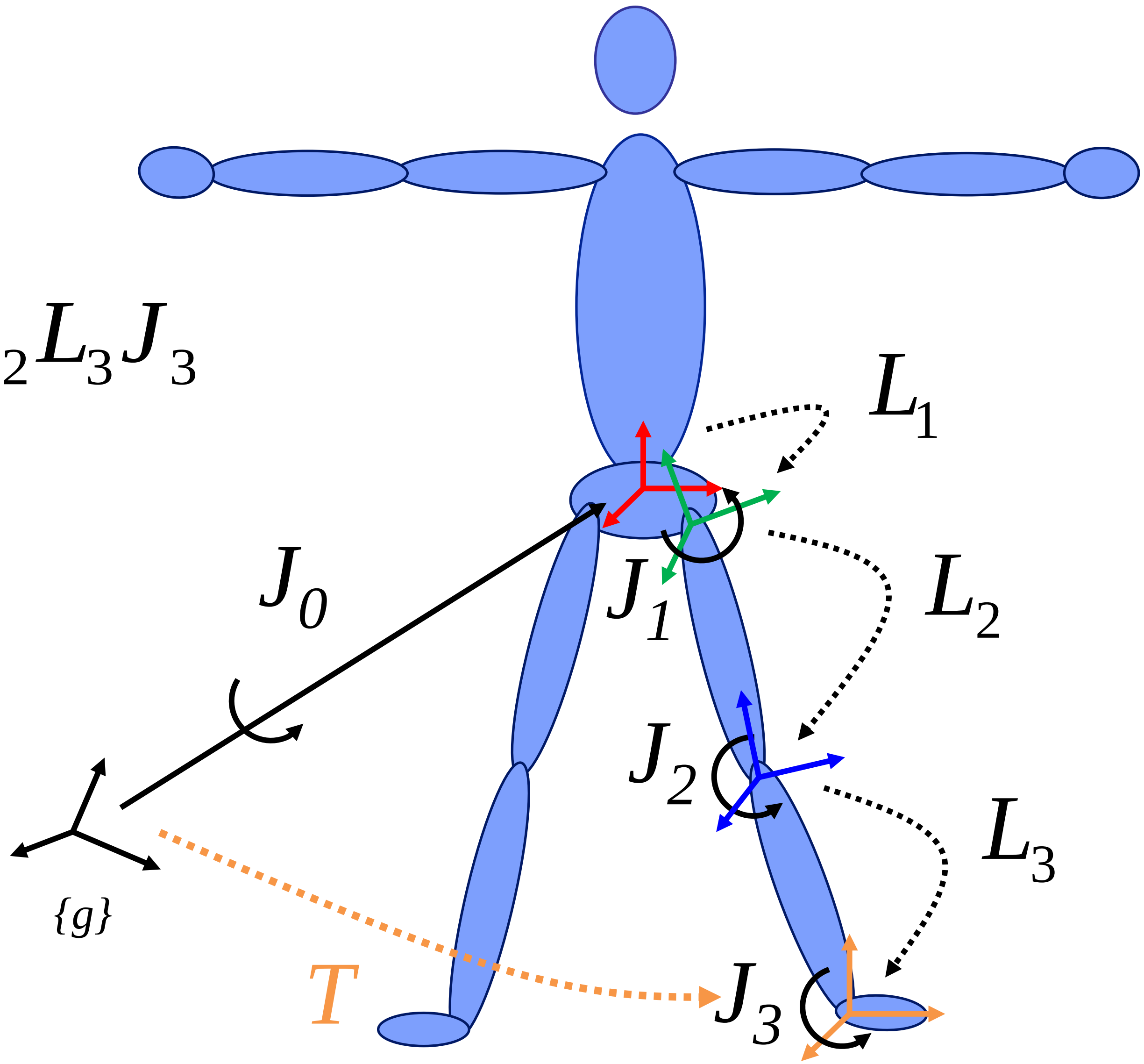
Forward Kinematics Map

$$T = J_0 L_1 J_1 L_2 J_2 L_3$$



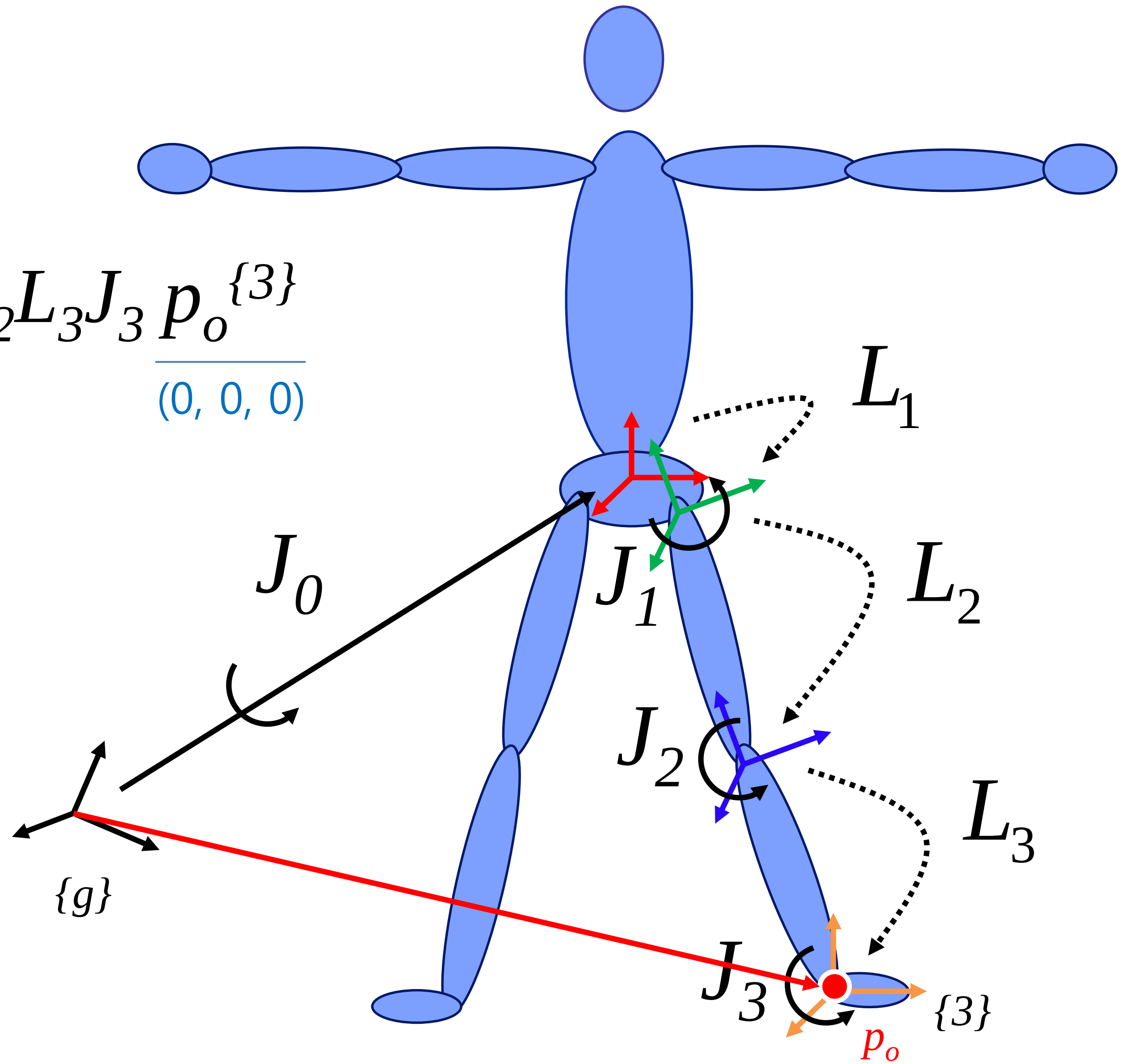
Forward Kinematics Map

$$T = J_0 L_1 J_1 L_2 J_2 L_3 J_3$$



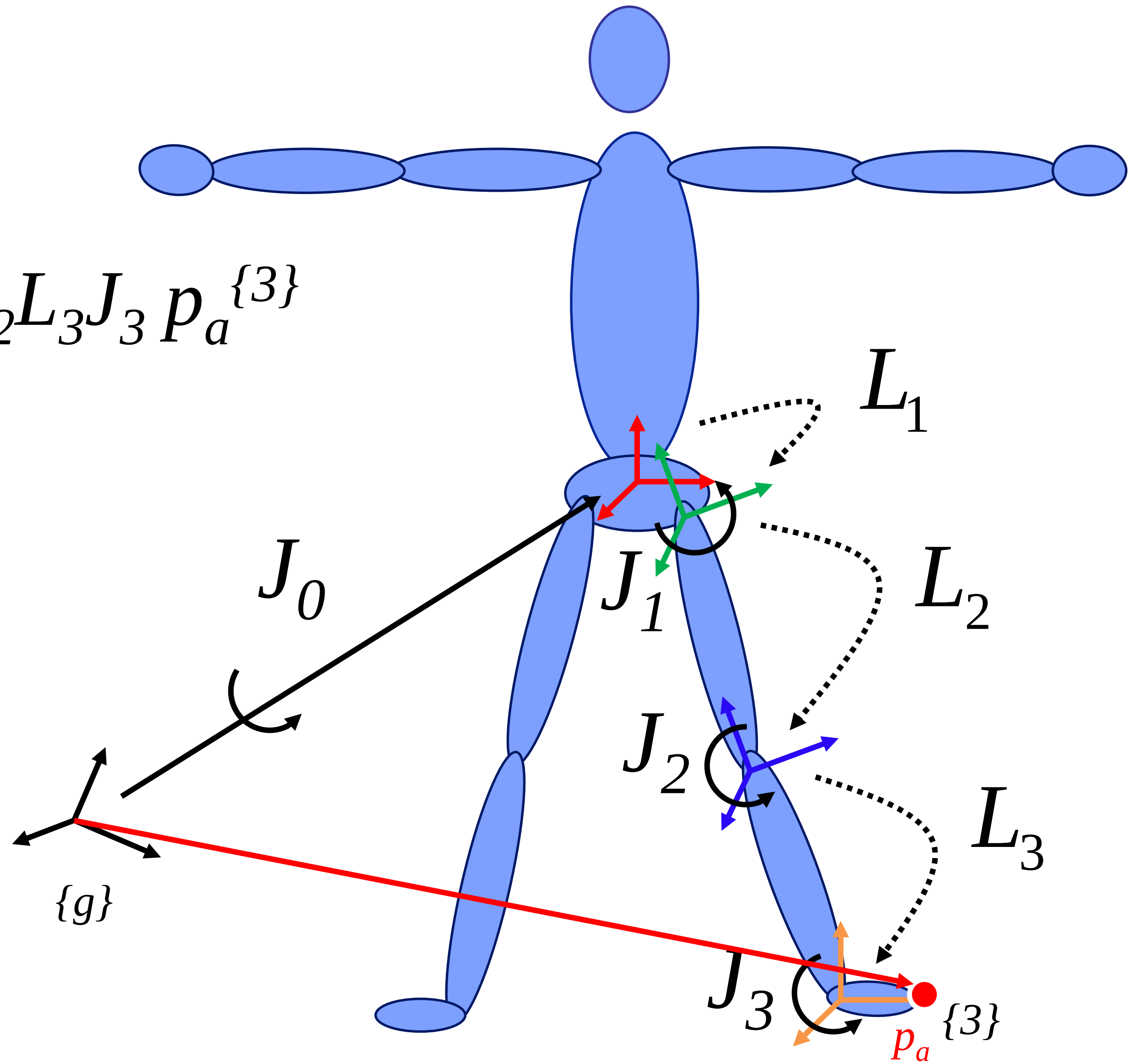
Forward Kinematics Map

$$p_o^{\{g\}} = J_0 L_1 J_1 L_2 J_2 L_3 J_3 \underbrace{p_o^{\{3\}}}_{(0,\,0,\,0)}$$



Forward Kinematics Map

$$p_a^{\{g\}} = J_0 L_1 J_1 L_2 J_2 L_3 J_3 \, p_a^{\{3\}}$$



Quiz #2

- Go to https://www.slido.com/
- Join #**cg-ys**
- Click “Polls”

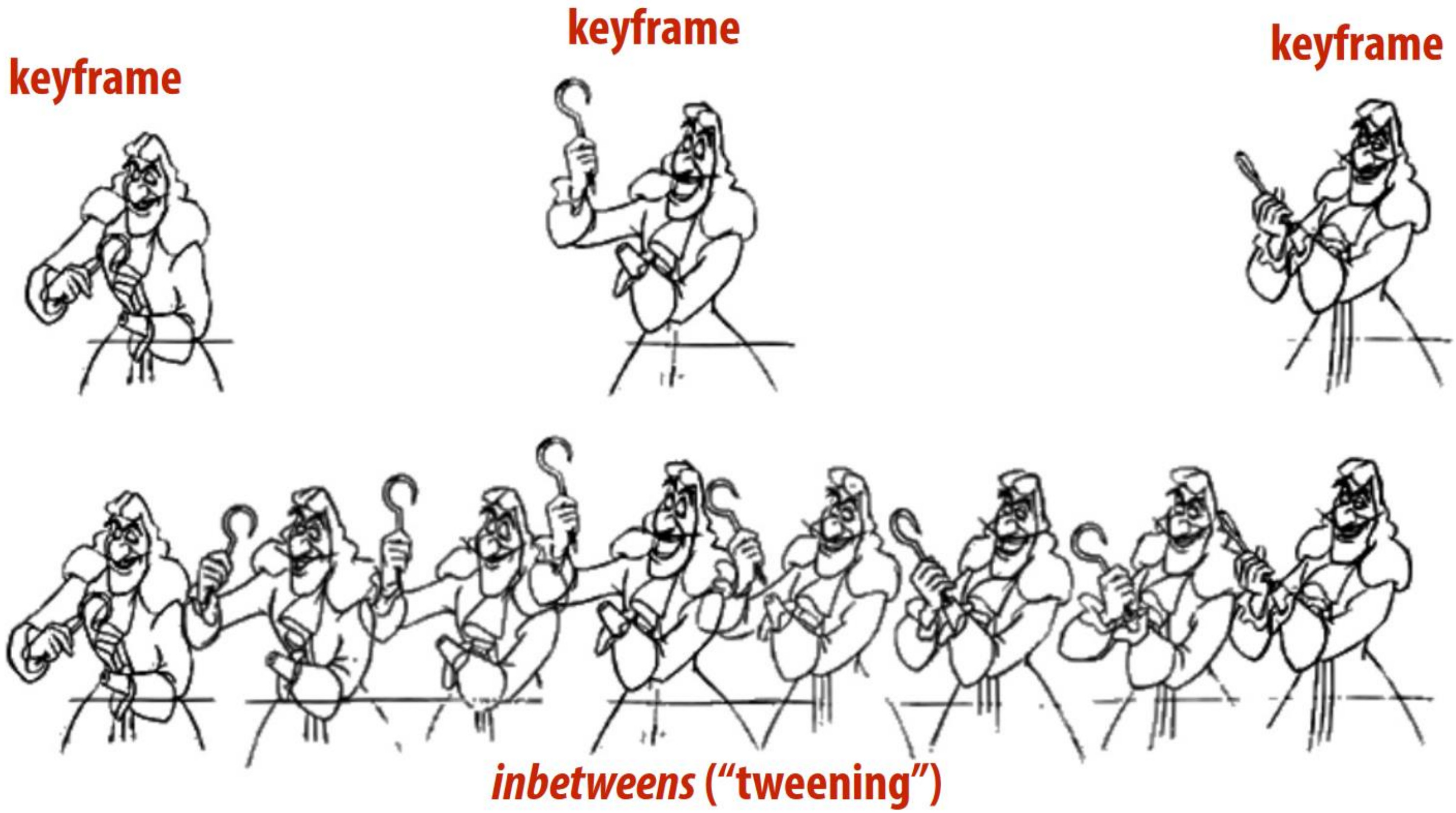
- Submit your answer in the following format:
  - **Student ID: Your answer**
  - **e.g. 2017123456: 4)**

- Note that you must submit all quiz answers in the above format to be checked for “attendance”.

Introduction to Character Animation

Traditional Hand-drawn Cel Animation

- Senior artist draws *keyframes*
- Assistant draws *inbetweens*
- Tedious / labor intensive (opportunity for technology!)



Animation by Milt Kahl (Walt Disney Studios)

Animation by Marc Davis (Walt Disney Studios)

Animation by Mark Henn (Walt Disney Studios)

Animation by Milt Kahl (Walt Disney Studios)

https://www.wnyc.org/story/sideshow-classic-disney-pencil-animations-come-life-gifs/

Computer Animation

- Computers are now widely replacing labor-intensive animation processes.
  - More controllable than drawing images by hands or constructing miniatures.

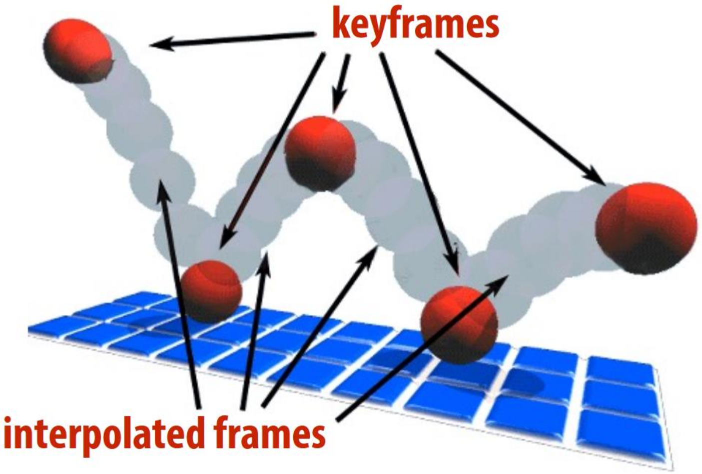
Character Animation Approaches

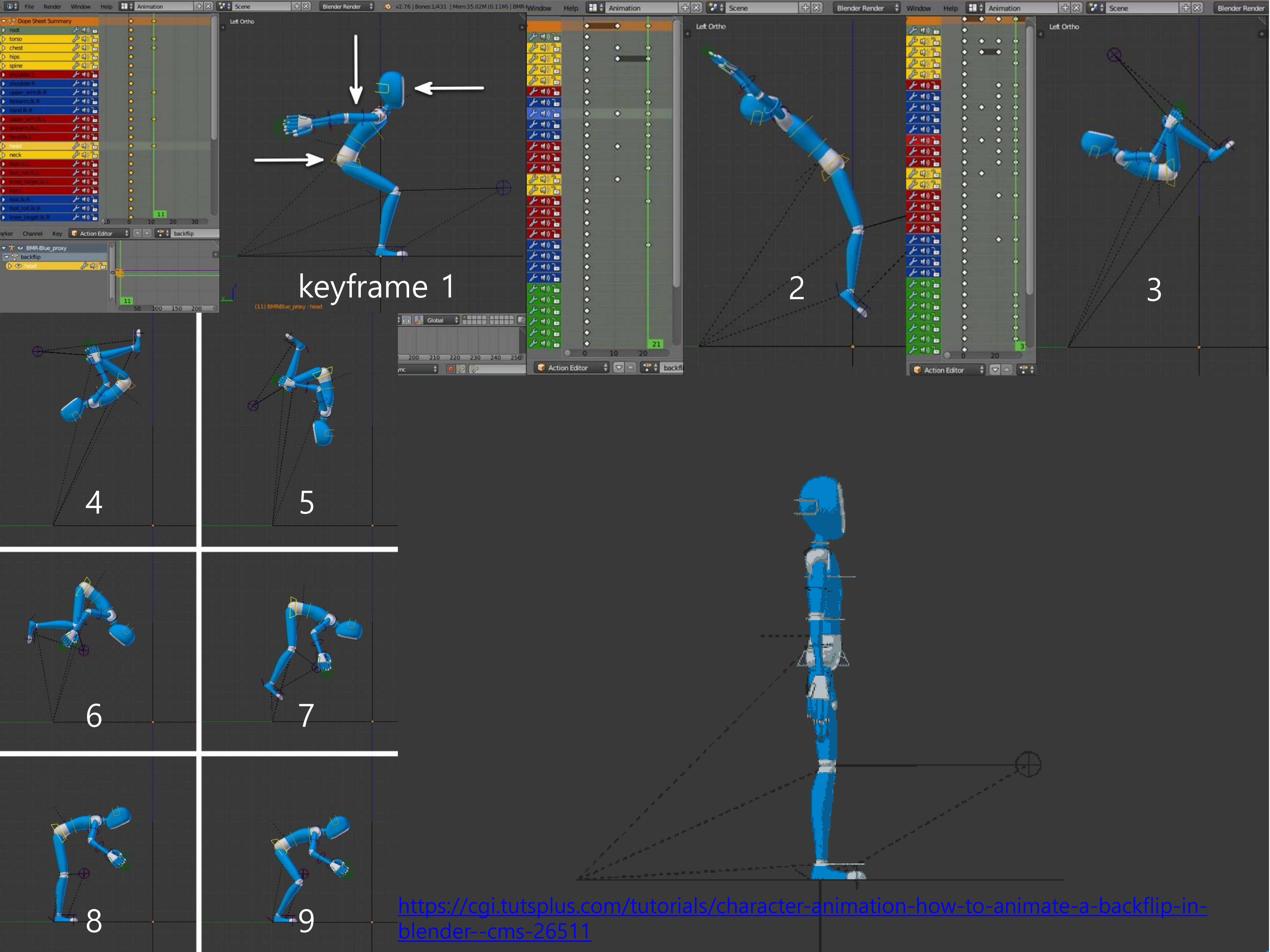
- Keyframe Animation
- Motion Capture
- Data-Driven Animation
  - Deep Motion Synthesis
- Physics-Based Animation
  - Learning Deep Control Policies

Keyframe Animation

- Idea:
  - Animators specifies important events at *key frames*.
  - Computer fills inbetween frames using interpolation.

- Events can be
  - Positions & orientation of objects
  - Light intensities
  - Camera parameters
  - ...





https://cgi.tutsplus.com/tutorials/character-animation-how-to-animate-a-backflip-in-blender--cms-26511

Keyframe Animation

- One of the earliest methods used to produce computer animation.

- Difficult to create “realistic” and “physically plausible” motions.
  - The quality of the output largely depends on the skill of the individual artist.

- Still used a lot.

Motion Capture

- Idea: Use “real” human motion to create realistic animation.

- Motion capture system “captures” movement of people or objects by measuring
  - position of each marker on the skin
  - position and orientation of each body part (or joint)

Motion Capture

https://youtu.be/YzS73UCOk20

Motion Capture

- Currently, widely used in movies & games
  - by major companies
- Very expansive
  - Expensive devices
  - High operating cost
- Limitation: Motion captured data is very realistic **only** in the same virtual environment as capture environment.

Data-Driven Animation

- Idea: Make the most of the motion capture data by
  - reusing mocap data with various motion editing techniques
  - or learning a machine learning model with mocap data
- to generate new motions.

- Recently, deep learning techniques are used a lot.
  - → Deep motion synthesis

Deep Motion Synthesis

- Learning a model generating new motions using
  - Discriminative models with FFNN, CNN, or RNN structures
  - Generative models with GAN, VAE, or Flow
  - Manifold learning with Autoencoders
  - Reinforcement learning to learn a policy
  - ...

Example

https://youtu.be/fTV7sXqO6ig

Harvey, Félix G., Mike Yurick, Derek Nowrouzezahrai, and Christopher Pal. “Robust Motion In-Betweening.” ACM Transactions on Graphics 39, no. 4 (SIGGRAPH 2020)

Physics-Based Animation

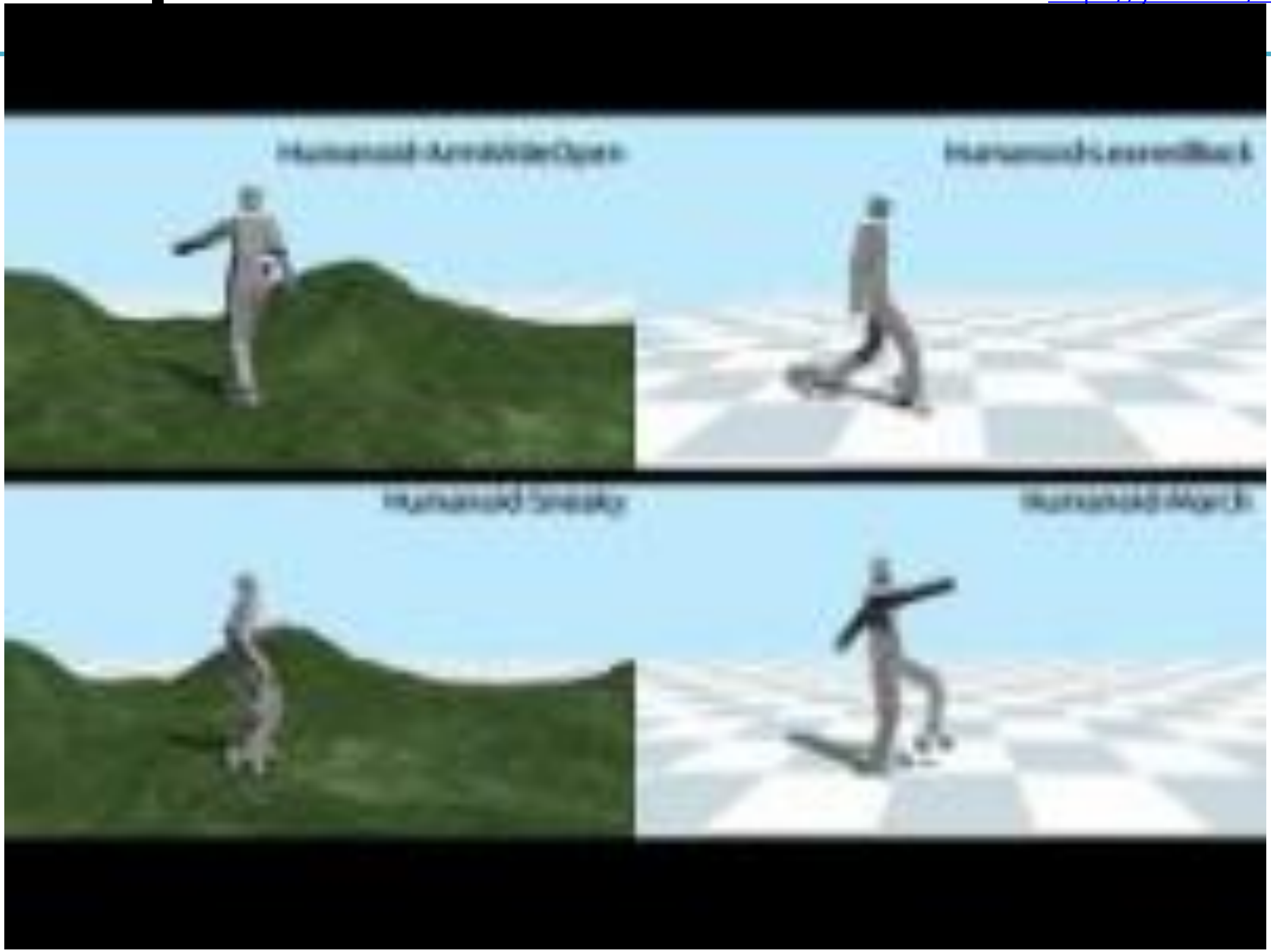
- Idea: Use physics simulation to generate motion
  - Because physical reality plays a key role in creating high-quality motion.
  - Physic simulation generates a motion that is always physically plausible.

- This requires a "controller".
  - Determines joint torques at each timestep to perform desired action while maintaining balance.
  - This problem is similar to that of robotics.

- Recently, deep reinforcement learning (DRL) is used a lot to learn deep control policies.

Example

https://youtu.be/fGwMrRoC5bw

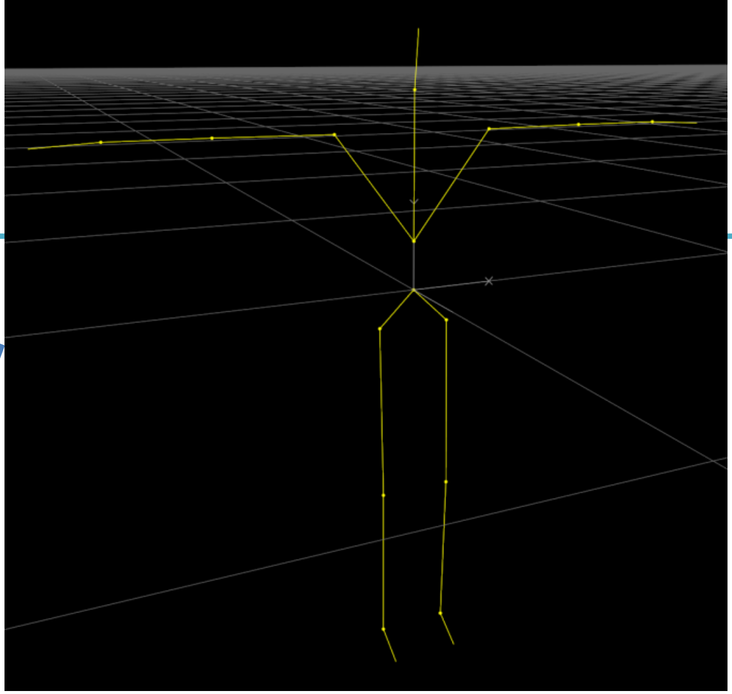


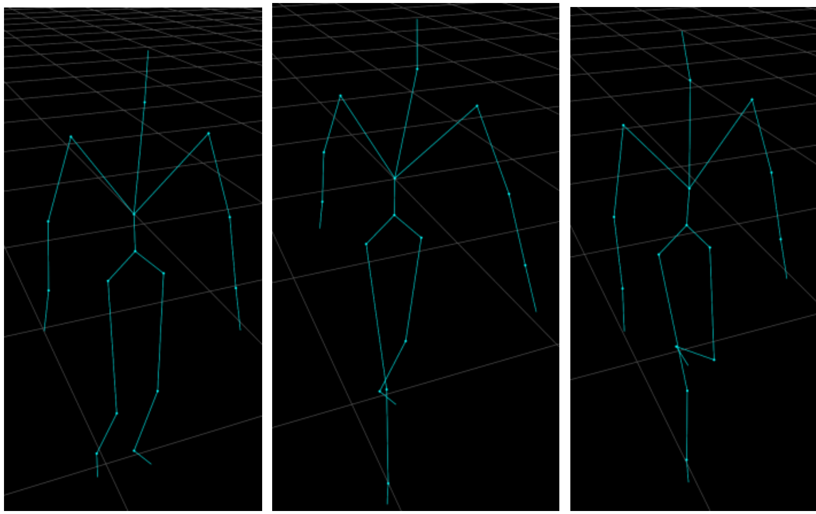
Kwon, Taesoo, Yoonsang Lee, and Michiel Van De Panne. “Fast and Flexible Multilegged Locomotion Using Learned Centroidal Dynamics.” ACM Transactions on Graphics 39, no. 4 (SIGGRAPH 2020)

Motion Capture Data

Motion Capture Data

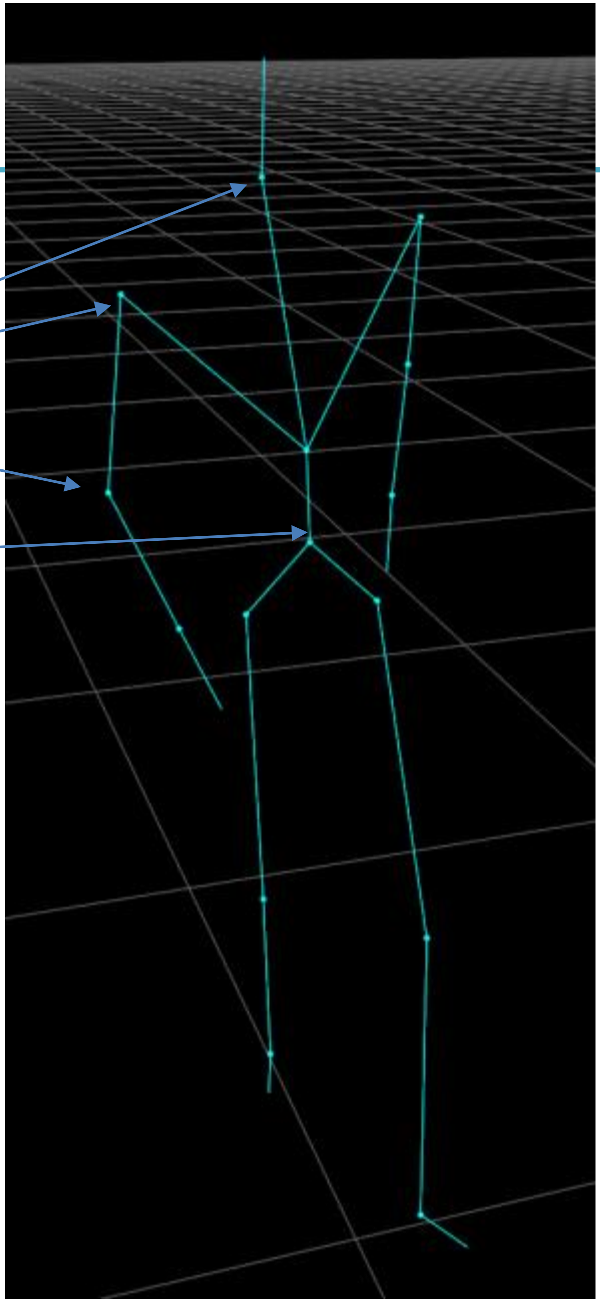
- Motion capture data includes two parts:
- "*Skeleton*": static data
  - joint hierarchy
  - joint offset from its parent joint - *link transformation Ls*
- "*Motion*": time-varying data
  - internal joint orientation (**w.r.t. default frame of each joint - the frame after applying link transformation for that joint**)
  - position and orientation of skeletal root (**w.r.t. global frame**)
  - → *joint transformations Js*
- *Posture* (*pose*): "motion" at a single frame
- **T pose**: a pose where all joint orientations are "zero" (identity matrix)



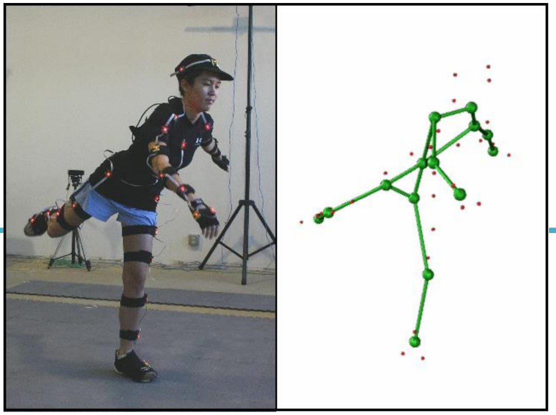


Motion Capture Data

- Generally, in motion capture data,
  - Internal joint has 3 DOFs
    - rotation only
  - Root joint has 6 DOFs
    - rotation and translation



BVH File Format



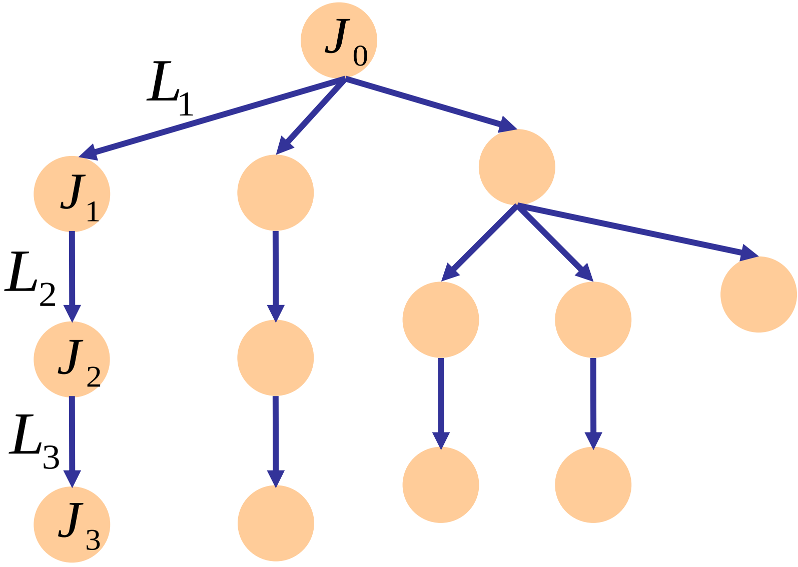
- BVH (**B**io**V**ision **H**ierarchical data)
  - Developed by Biovision, a motion capture company

- Consists of two parts:
- **Hierarchy section**
  - Describes the "*Skeleton*": static data
- **Motion section**
  - Describes the "*Motion*": time-varying data

- Text file format (human-readable)

Hierarchy Section

- The hierarchy is a joint tree.
- Each joint has
  - *Offset* from its parent joint - *link transformation L*
  - *Channel list*, which defines the type of its *joint transformation J*



Biovision BVH

- **Hierarchy section**
  - **"HIERARCHY“**
    - **"ROOT"**
      - followed by the name of the root
      - **" { "** and **" } "** pair
      - **"OFFSET"**
        - » X,Y and Z offset of the segment from its parent
      - **"CHANNELS"**
        - » the number of channels
        - » the type of each channel
    - **"JOINT"**
      - identical to the root definition except for the number of channels
      - **"OFFSET " , "CHANNELS"**
    - **"End Site"**
      - **indicates that the current segment is an end effector** (no children)
      - **"OFFSET "**

```
HIERARCHY
ROOT Hips
{
    OFFSET  0.00    0.00    0.00
    CHANNELS 6 Xposition Yposition Zposition Zrotation Xrotation Yrotation
    JOINT LeftHip
    {
        OFFSET   3.430000        0.000000    0.000000
        CHANNELS 3 Zrotation Xrotation Yrotation
        JOINT LeftKnee
        {
            OFFSET  0.000000  -18.469999  0.000000
            CHANNELS 3 Zrotation Xrotation Yrotation
            JOINT LeftAnkle
            {
                OFFSET      0.000000  -17.950001  0.000000
                CHANNELS 3 Zrotation Xrotation Yrotation
                End Site
                {
                    OFFSET      0.000000  -3.119999  0.000000
                }
            }
        }
    }
...
}
```

in this example,

- 6 channels for the root (Tx Ty Tz Rz Rx Ry)

- 3 channels for every other object (Rz Rx Ry)

```
HIERARCHY
ROOT Hips
{
  OFFSET 0.0 0.0 0.0
  CHANNELS 6 XPOSITION YPOSITION ZPOSITION ZROTATION XROTATION YROTAT
    JOINT chest
    {
            OFFSET  0.096536 3.475309 -0.289609
            CHANNELS 3 Xrotation Zrotation Yrotation
            JOINT neck
            {
                    OFFSET  -0.096536 13.901242 -2.027265
                    CHANNELS 3 Xrotation Zrotation Yrotation
                    JOINT head
                    {
                            OFFSET  -0.166775 1.448045 0.482682
                            CHANNELS 3 Xrotation Zrotation Yrotation
                            JOINT leftEye
```

```
HIERARCHY
ROOT Hips
{
  OFFSET 0.0 0.0 0.0                                              J0 channels
  CHANNELS 6 XPOSITION YPOSITION ZPOSITION ZROTATION XROTATION YROTAT
    JOINT chest
    {
            OFFSET  0.096536 3.475309 -0.289609 L1
            CHANNELS 3 Xrotation Zrotation Yrotation J1 channels
            JOINT neck
            {
                    OFFSET  -0.096536 13.901242 -2.027265 L2
                    CHANNELS 3 Xrotation Zrotation Yrotation J2 channels
                    JOINT head
                    {
                            OFFSET  -0.166775 1.448045 0.482682 L3
                            CHANNELS 3 Xrotation Zrotation Yrotation
                            JOINT leftEye                     J3 channels
```

```
HIERARCHY
ROOT Hips
{
  OFFSET 0.0 0.0 0.0
  CHANNELS 6 XPOSITION YPOSITION ZPOSITION ZROTATION XROTATION YROTAT
    JOINT chest
    {
          OFFSET  0.096536 3.475309 -0.289609
          CHANNELS 3 Xrotation Zrotation Yrotation
          JOINT neck
          {
                OFFSET  -0.096536 13.901242 -2.027265
                CHANNELS 3 Xrotation Zrotation Yrotation
                JOINT head
                {
                      OFFSET  -0.166775 1.448045 0.482682
                      CHANNELS 3 Xrotation Zrotation Yrotation
                      JOINT leftEye
```

Root Hips Joint

Root offset is generally zero (or ignored even if it's not zero)

Chest Joint

Neck Joint

Neck's offset from chest

Channel list:
Transformation from chest frame to neck frame

Biovision BVH

```
MOTION
Frames:     20
Frame Time: 0.033333
 0.00  39.68   0.00    0.65   ...
...
```

■ Motion Section

➢ "MOTION"

- followed by a line indicating the number of frames
- **"Frames:"**
  - the number of frames
- **"Frame Time:"**
  - the sampling rate of the data
  - Ex) 0.033333 ➔ 30 frames a second
- The rest of the file contains the actual motion data
  - The numbers appear in the order of the channel specifications as the skeleton hierarchy was parsed

- **Each line has motion data for a single frame**
- **Each number in a line is a value for a single channel**
- **The unit of rotation channel values is degree**

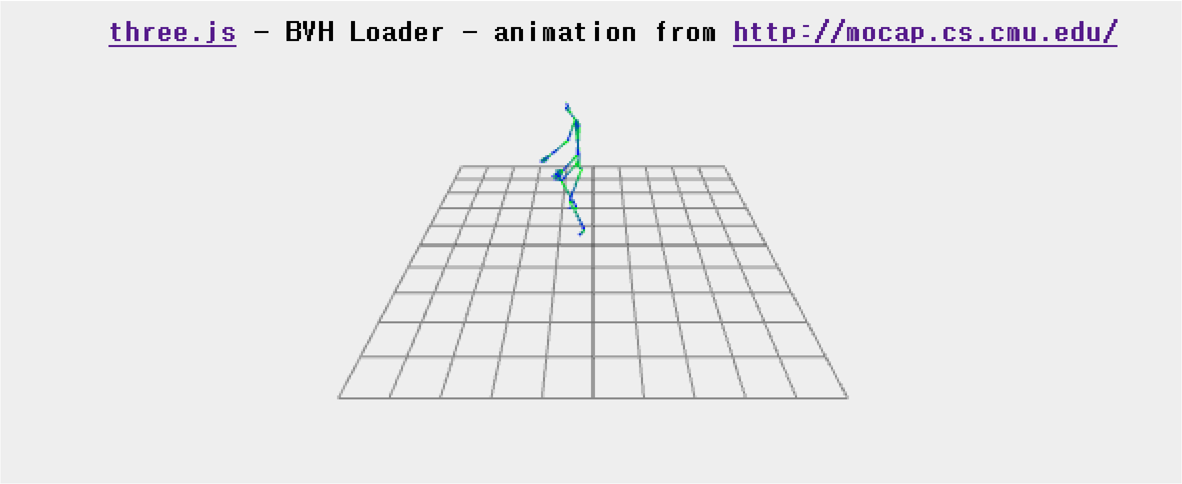
```
HIERARCHY
ROOT Hips
{
  OFFSET 0.0 0.0 0.0
  CHANNELS 6 XPOSITION YPOSITION ZPOSITION ZROTATION XROTATION YROTATION
    JOINT chest Column 1     Column 2     Column 3     Column 4     Column 5     Column 6
    {
          OFFSET  0.096536 3.475309 -0.289609
          CHANNELS 3 Xrotation Zrotation Yrotation
          JOINT neck    Column 7    Column 8    Column 9
          {
                OFFSET  -0.096536 13.901242 -2.027265
                CHANNELS 3 Xrotation Zrotation Yrotation
                JOINT head   Column 10   Column 11   Column 12
                {
                      OFFSET  -0.166775 1.448045 0.482682
                      CHANNELS 3 Xrotation Zrotation Yrotation
                                 Column 13    Column 14    Column 15
MOTION
Frames: 199
Frame Time: 0.033333
1.95769 0.989769479321 0.039193 -4.11275998891 -0.490682977769 -91.3519974695 0.45458697547  ...
1.95769 0.989769479321 0.0392908 -4.11760985011 -0.48626597981 -91.3734989051 0.513819016282  ...
1.95769 0.989769479321 0.039424 -4.12004011679 -0.488125979059 -91.387002189 0.592700017233  ...
1.95771 0.989769479321 0.0395518 -4.0961698863 -0.500940000911 -91.3840993586 0.61126399115  ...
1.95779 0.989759479321 0.0396999 -4.05799980101 -0.510696019006 -91.3839969058 0.58299101005  ...
1.9579 0.989719479321 0.0398625 -4.0423300664 -0.503295989288 -91.3842018115 0.57718001317  ...
...
```

Biovision BVH

Interpreting the data

- To calculate the position of a segment
  - Translation information
    - For any joint segment
      - the translation information will simply be the offset as defined in the hierarchy section
    - For the root object
      - The translation data will be the sum of the offset data and the translation data from the motion section
  - Rotation information
    - comes from the motion section
  - **The "CHANNELS" order is important: If the order is “ZROTATION XROTATION YROTATION”**
    - **Apply transformation in order of rotation about z, rotation about x, rotation about y w.r.t. local frame**
    - **→ ZXY Euler angles**
  - **Do not assume ZXY Euler angles. Other sequences may also be used.**

[Practice] BVH Online Demo



http://motion.hahasoha.net/

- Select other motions from the list.
- Download corresponding BVH files and open them in a text editor.

Quiz #3

- Go to https://www.slido.com/
- Join #**cg-ys**
- Click “Polls”

- Submit your answer in the following format:
  - **Student ID: Your answer**
  - **e.g. 2017123456: 4)**

- Note that you must submit all quiz answers in the above format to be checked for “attendance”.

Next Time

- Lab for this lecture (next Monday):
  - Lab assignment 10

- Next lecture:
  - 11 - Curve

- **Class Assignment #3**
  - **Due: 23:59, Jun 7, 2021**

- Acknowledgement: Some materials come from the lecture slides of
  - Prof. Kayvon Fatahalian and Prof. Keenan Crane, CMU, http://15462.courses.cs.cmu.edu/fall2015/
  - Prof. Jinxiang Chai, Texas A&M Univ., http://faculty.cs.tamu.edu/jchai/csce441_2016spring/lectures.html
  - Prof. Jehee Lee, SNU, http://mrl.snu.ac.kr/courses/CourseGraphics/index_2017spring.html
  - Prof. Taesoo Kwon, Hanyang Univ., http://calab.hanyang.ac.kr/cgi-bin/cg.cgi