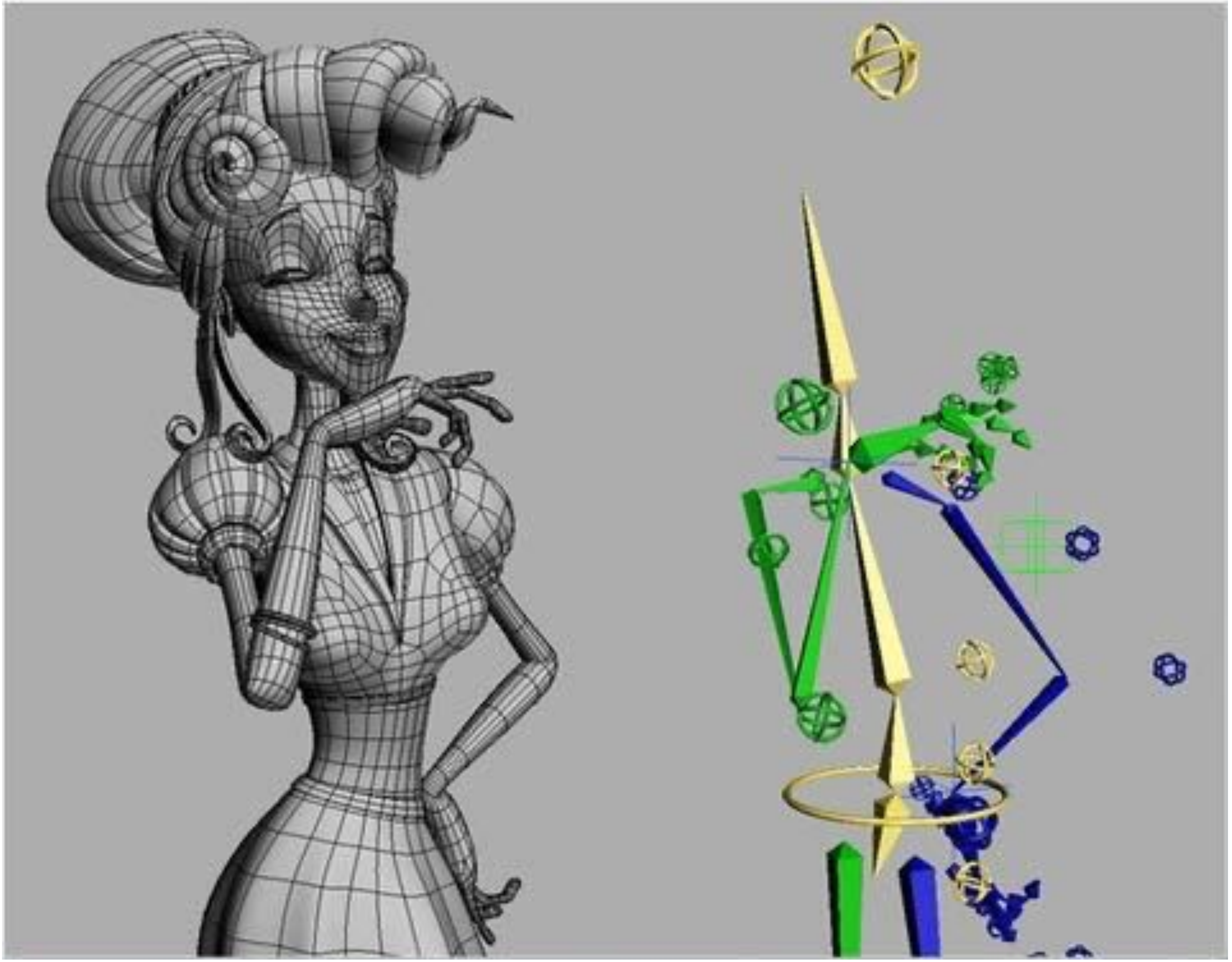
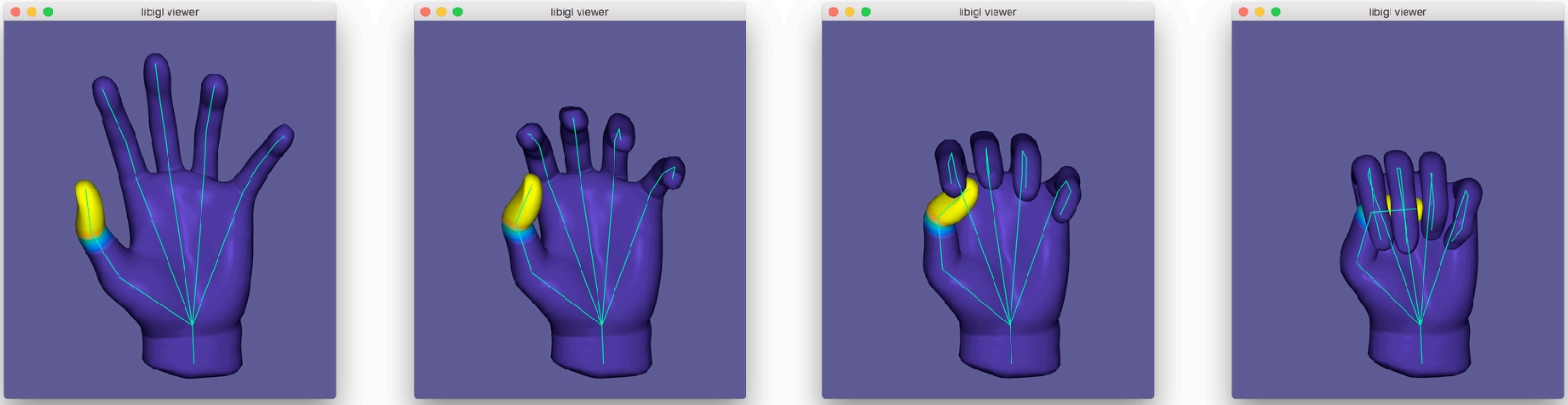
COL781: Computer Graphics

31. Skeletal Animation

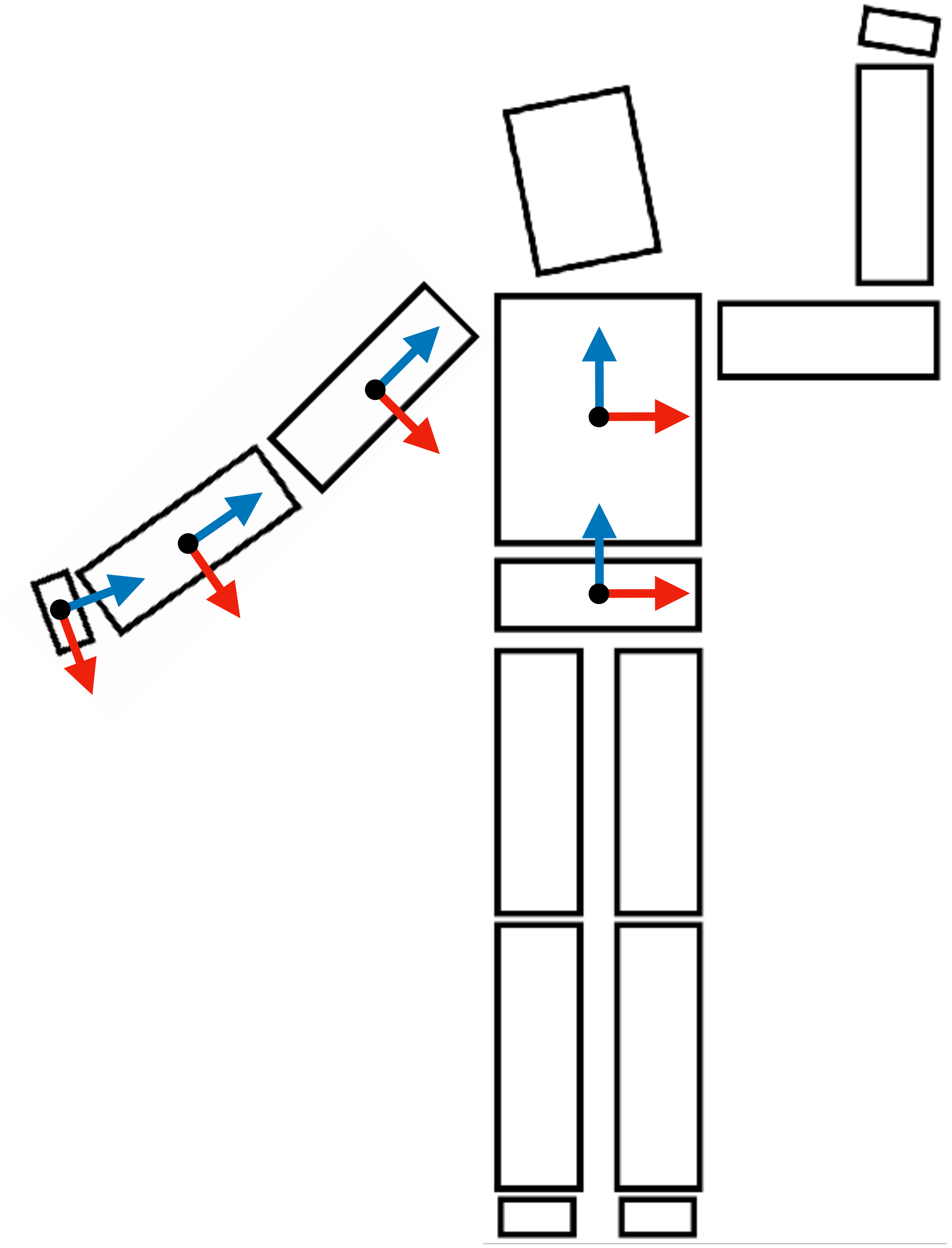
Pixar



Skeletal motion: transformation of bones based on joint angles



Skinning: displacement of surface vertices based on bone transformations



Recall hierarchical transformations from Lec. 6!

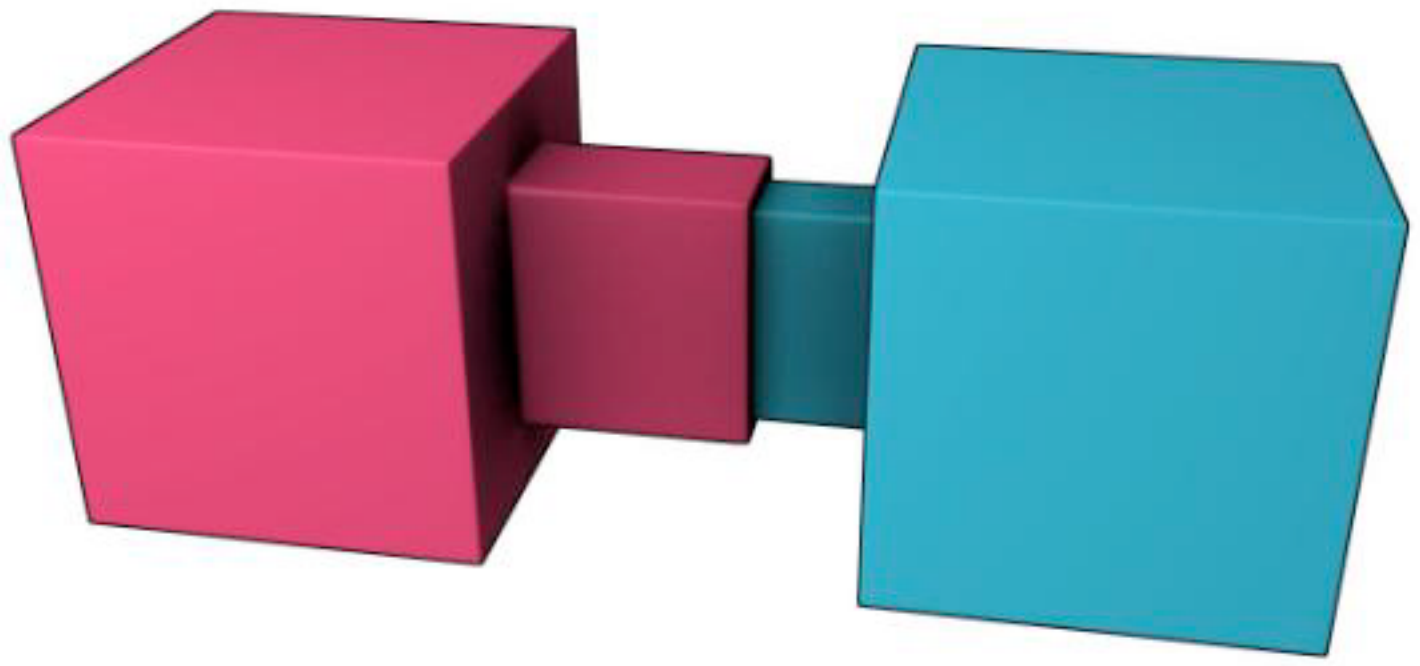
Each bone is transformed relative to its parent

$$\mathbf{M}_{\text{lHand}}^{\text{world}} = \mathbf{M}_{\text{hip}}^{\text{world}} \mathbf{M}_{\text{chest}}^{\text{hip}} \mathbf{M}_{\text{ulArm}}^{\text{chest}} \mathbf{M}_{\text{llArm}}^{\text{ulArm}} \mathbf{M}_{\text{lHand}}^{\text{llArm}}$$

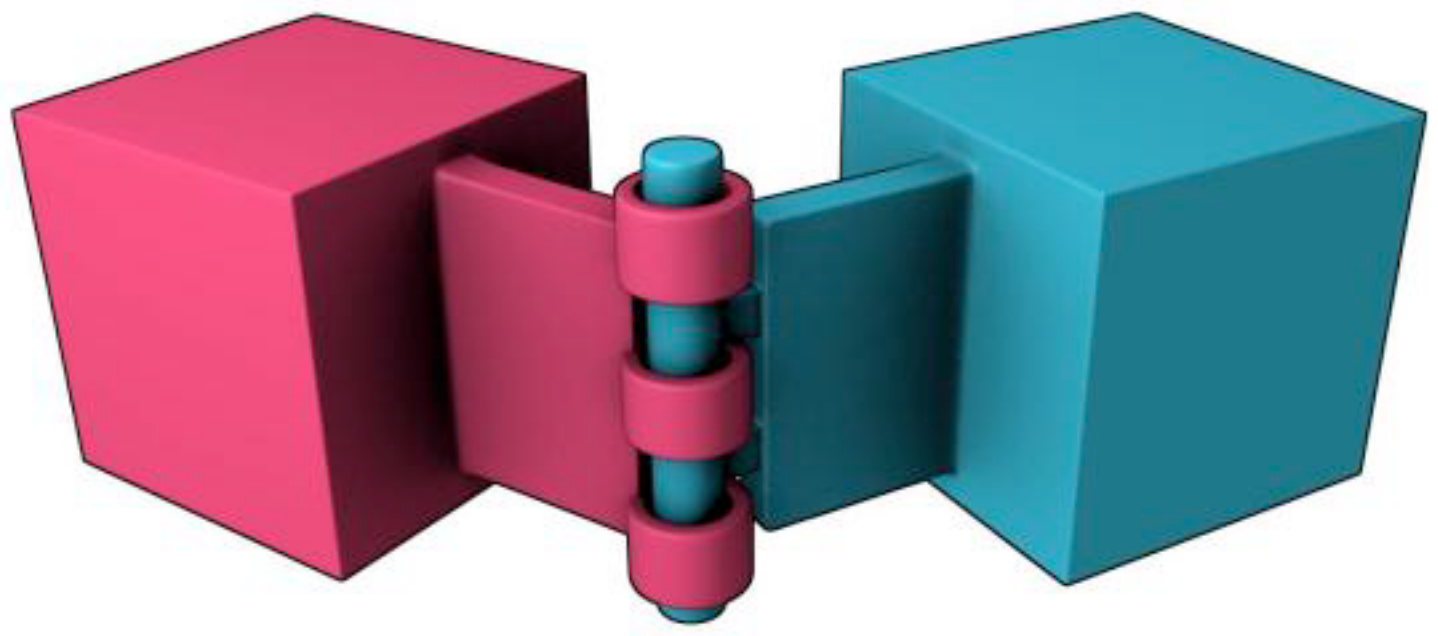
$$\mathbf{M}_{\text{lHand}}^{\text{world}} = \mathbf{M}_{\text{llArm}}^{\text{world}} \mathbf{M}_{\text{lHand}}^{\text{llArm}}$$

OK, how to control transformation of child bone?

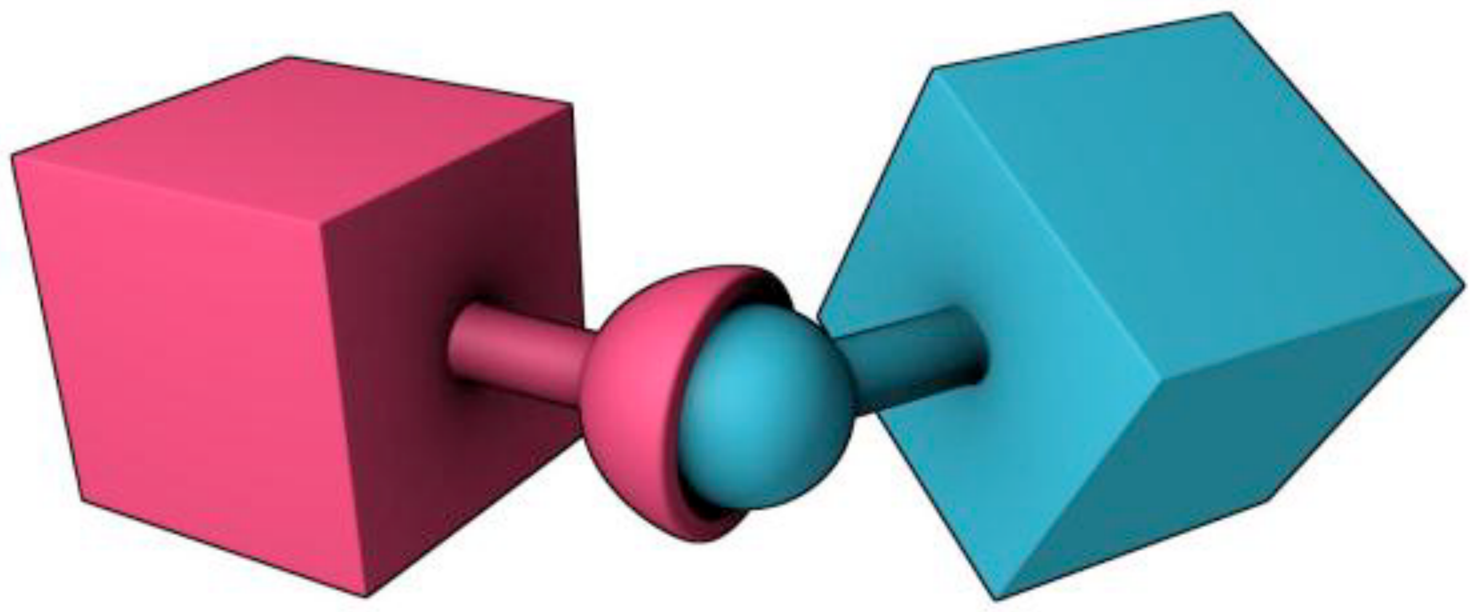
$\mathbf{M}_{\text{child}}^{\text{parent}}$ cannot be arbitrary! Child's relative motion is constrained by some kind of **joint**, e.g.:



Slider
(1 DOF)



Hinge
(1 DOF)



Ball-and-socket
(3 DOFs)

Bullet manual

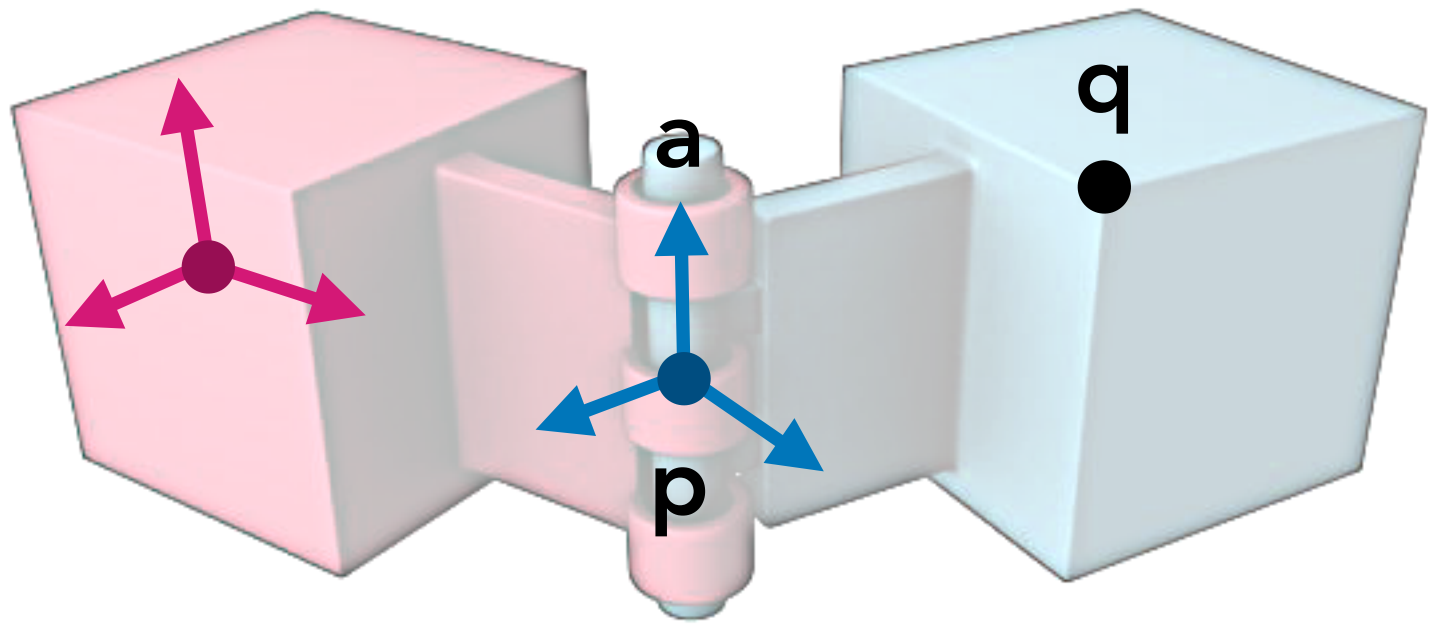
(Lots of other types of joints in mechanical engineering…)

Example: Hinge joint

Suppose hinge point is at **p** in parent's coordinates, origin in child's coordinates

Hinge axis vector is **a** in both coordinate systems (why?)

Consider a point at coordinates $\mathbf{q}^c$ on the child bone.



- After rotation about hinge axis: $\mathbf{R}(\theta, \mathbf{a})\ \mathbf{q}^c$
- In parent's coordinate system: $\mathbf{T}(\mathbf{p})\ \mathbf{R}(\theta, \mathbf{a})\ \mathbf{q}^c$

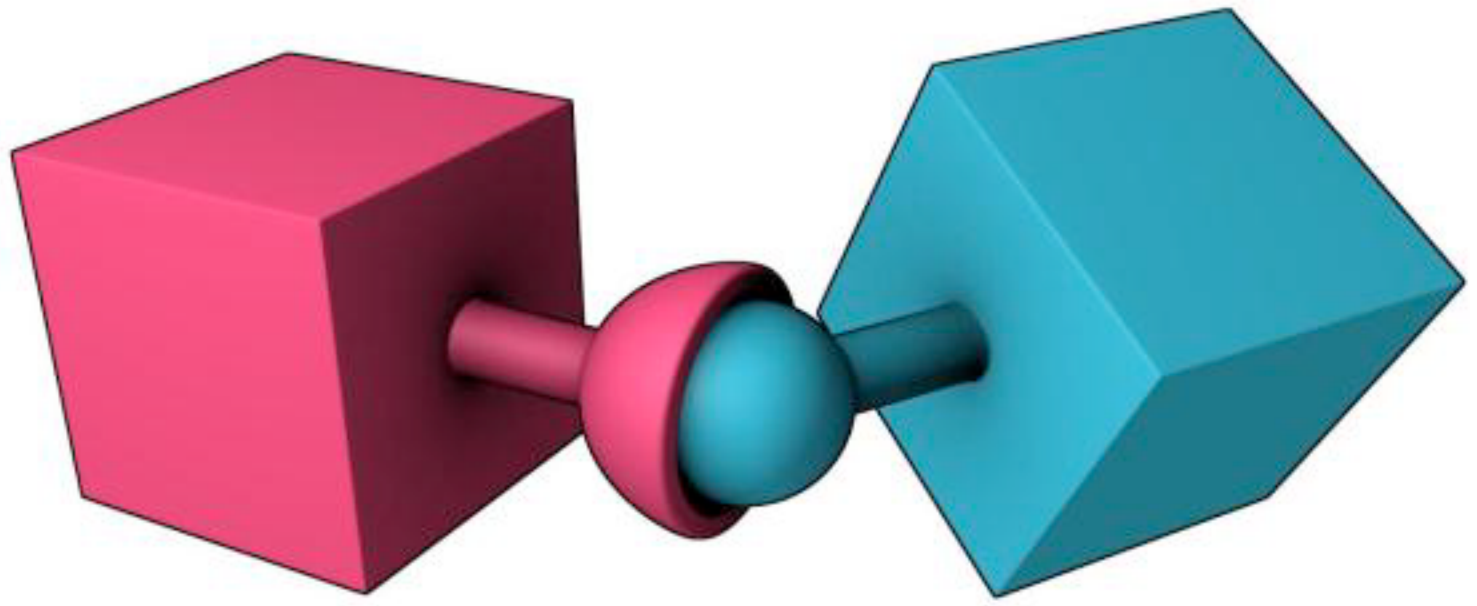
Full transformation:

$$\mathbf{q}^p = \mathbf{T}(\mathbf{p})\ \mathbf{R}(\theta, \mathbf{a})\ \mathbf{q}^c$$

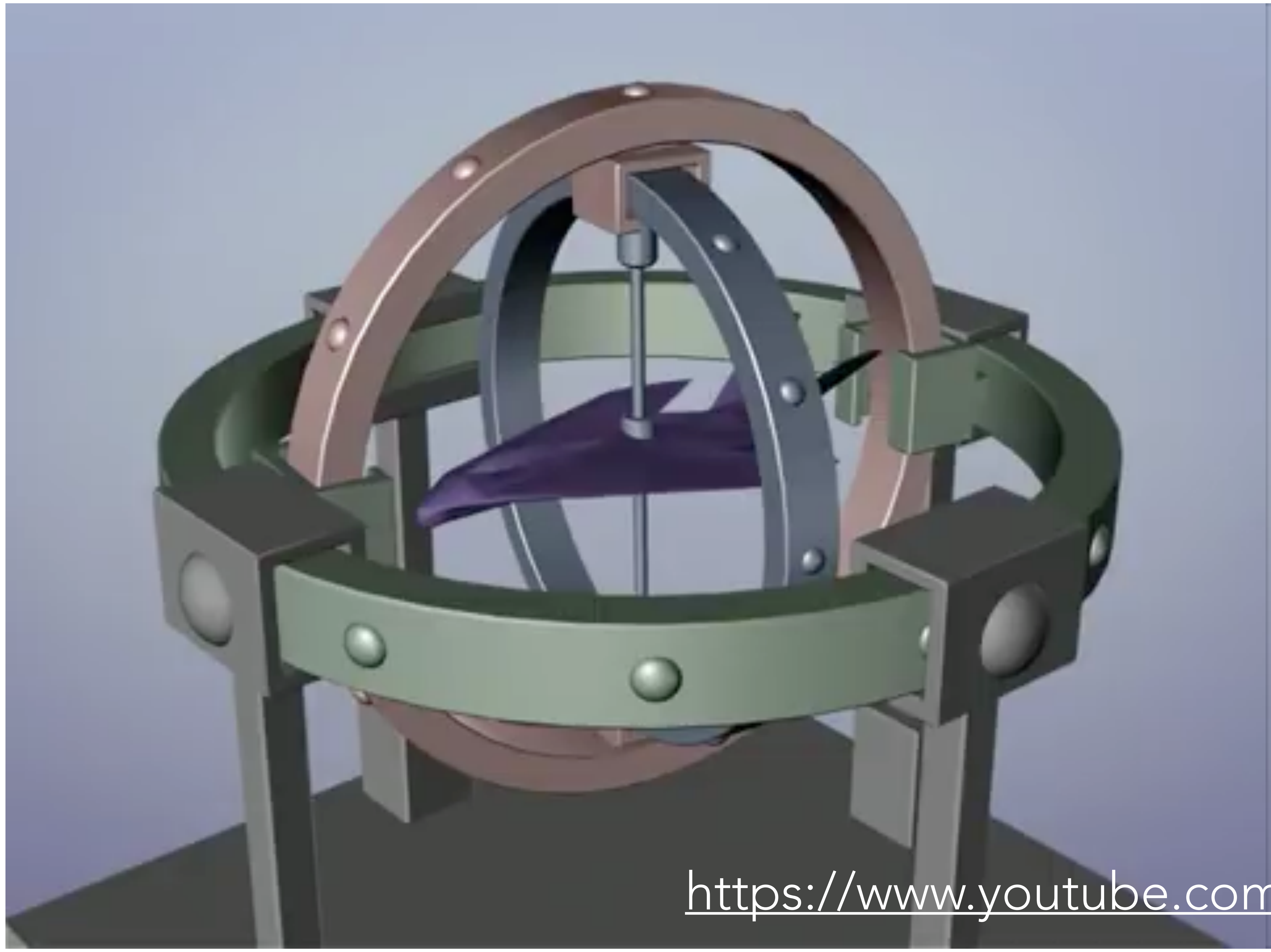
$$\mathbf{M}_c^p = \mathbf{T}(\mathbf{p})\ \mathbf{R}(\theta, \mathbf{a})$$

How to represent the rotation of a ball joint?

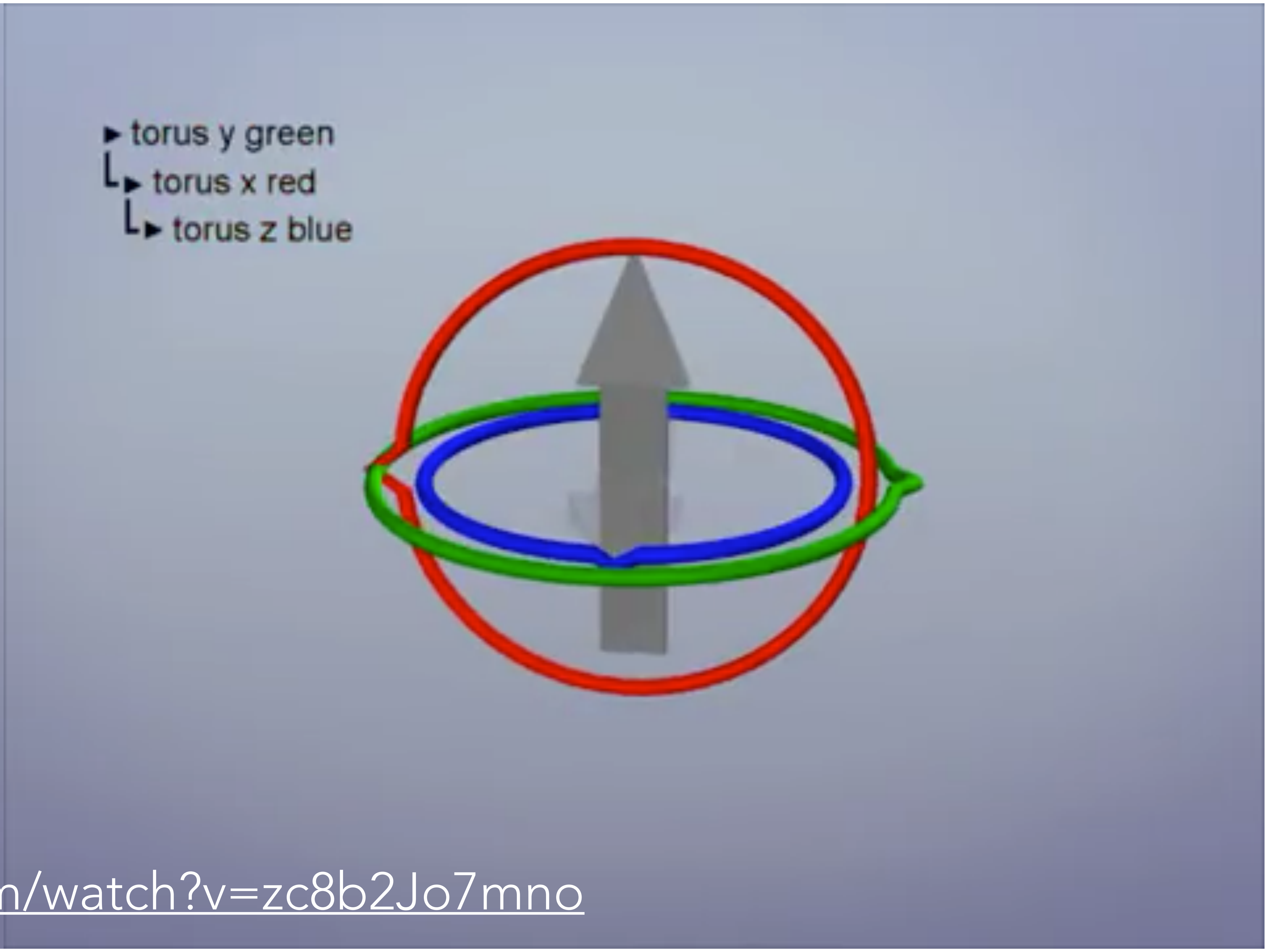
- 3×3 rotation matrix $\mathbf{R}$?
- Euler angles $(\theta_x, \theta_y, \theta_z)$?
- Quaternions $\mathbf{q} = a + bi + cj + dk$?



Problems with Euler angles



Gimbal lock



Unnatural interpolation

https://www.youtube.com/watch?v=zc8b2Jo7mno

GuerillaCG

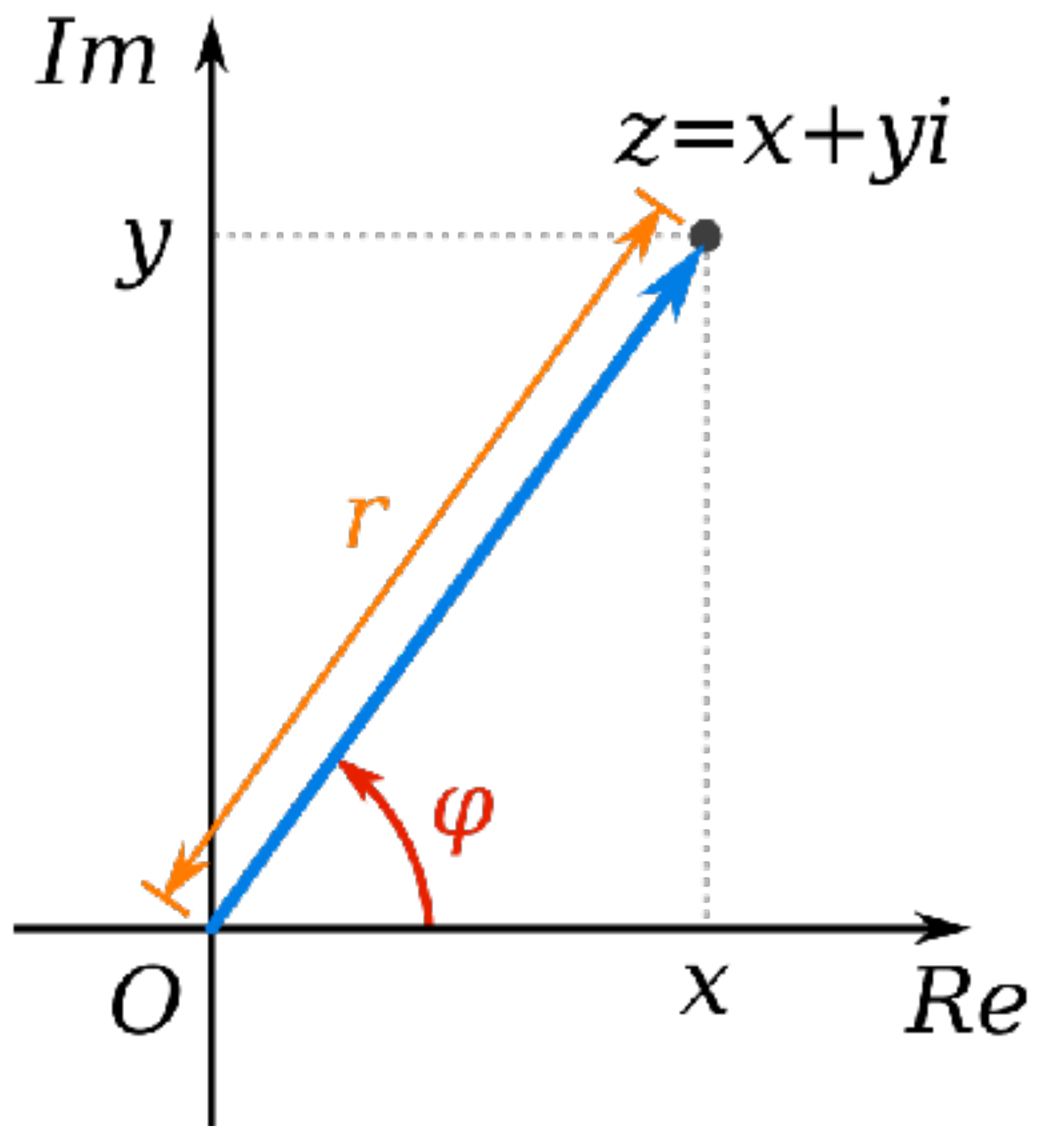
Warm-up: 2D rotations via complex numbers

Complex numbers are quantities of the form $a + ib$ where $i^2 = -1$.

Any unit complex number ($a^2 + b^2 = 1$) represents a 2D rotation.
Rotation by angle θ:

$$z = \cos(\theta) + i \sin(\theta)$$

How to rotate a vector $\mathbf{v} = (x, y) \in \mathbb{R}^2$? Interpret it as a complex number $v = x + iy$. Then the rotated vector is zv.



Quaternions

Quaternions are quantities of the form $\mathbf{q} = a + b\mathbf{i} + c\mathbf{j} + d\mathbf{k}$ where

$$\mathbf{i}^2 = \mathbf{j}^2 = \mathbf{k}^2 = \mathbf{ijk} = -1$$

Useful to separate into scalar part and vector part: $\mathbf{q} = (a, \mathbf{u})$

- Multiplication: Not commutative! $\mathbf{q}_1 \mathbf{q}_2 \neq \mathbf{q}_2 \mathbf{q}_1$ in general

$$(a, \mathbf{u})(b, \mathbf{v}) = (ab - \mathbf{u} \cdot \mathbf{v}, a\mathbf{v} + b\mathbf{u} + \mathbf{u} \times \mathbf{v})$$

- Conjugate: $\mathbf{q}^* = a - b\mathbf{i} - c\mathbf{j} - d\mathbf{k} = (a, -\mathbf{u})$.
- Norm: $|\mathbf{q}| = \sqrt{\mathbf{q}^*\mathbf{q}} = \sqrt{a^2 + b^2 + c^2 + d^2}$

	1	**i**	**j**	**k**
1	1	**i**	**j**	**k**
i	**i**	−1	**k**	−**j**
j	**j**	−**k**	−1	**i**
k	**k**	**j**	−**i**	−1

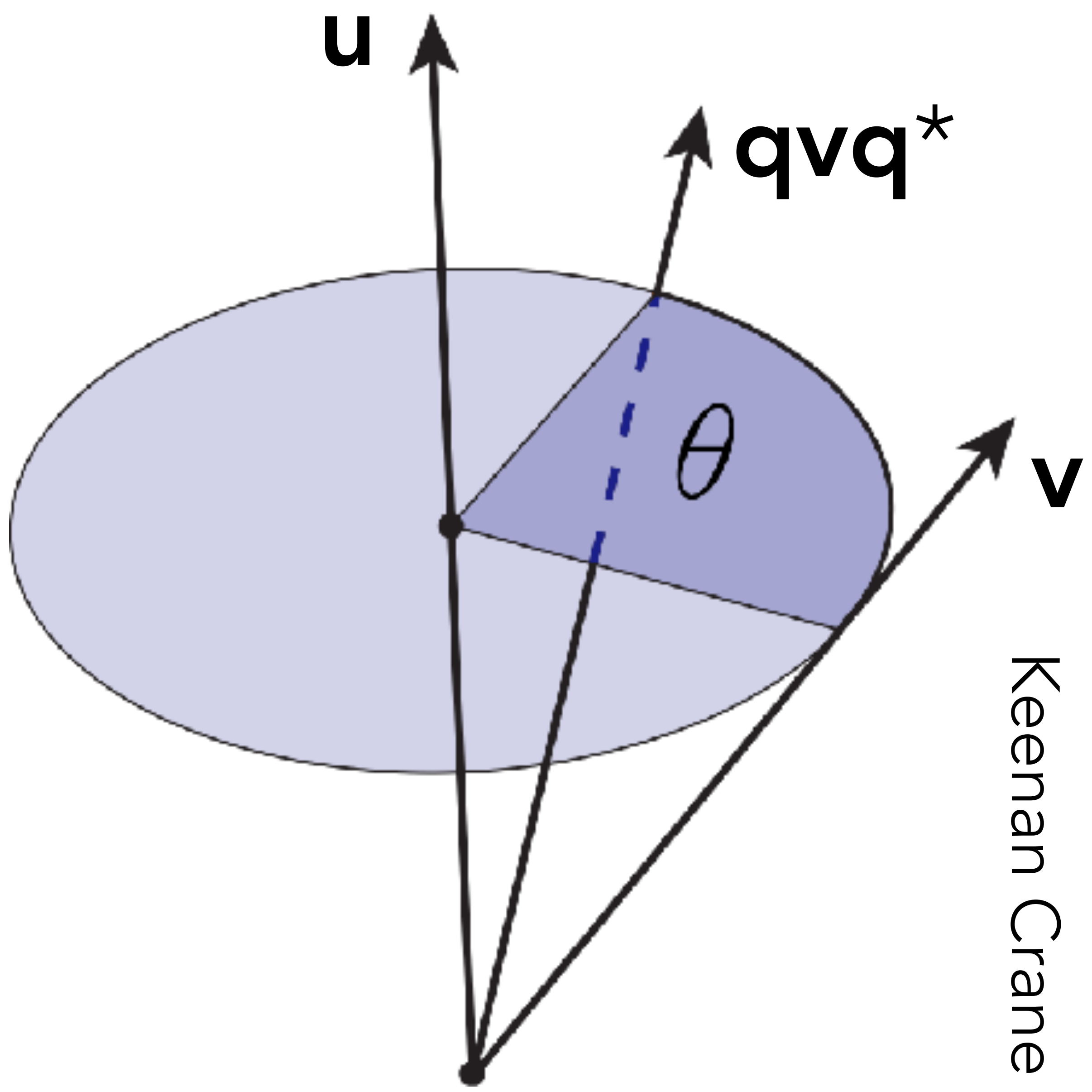
Any unit quaternion represents a 3D rotation. Rotation by angle θ about axis $\mathbf{u}$:

$$\mathbf{q} = (\cos(\theta/2),\ \mathbf{u}\sin(\theta/2))$$

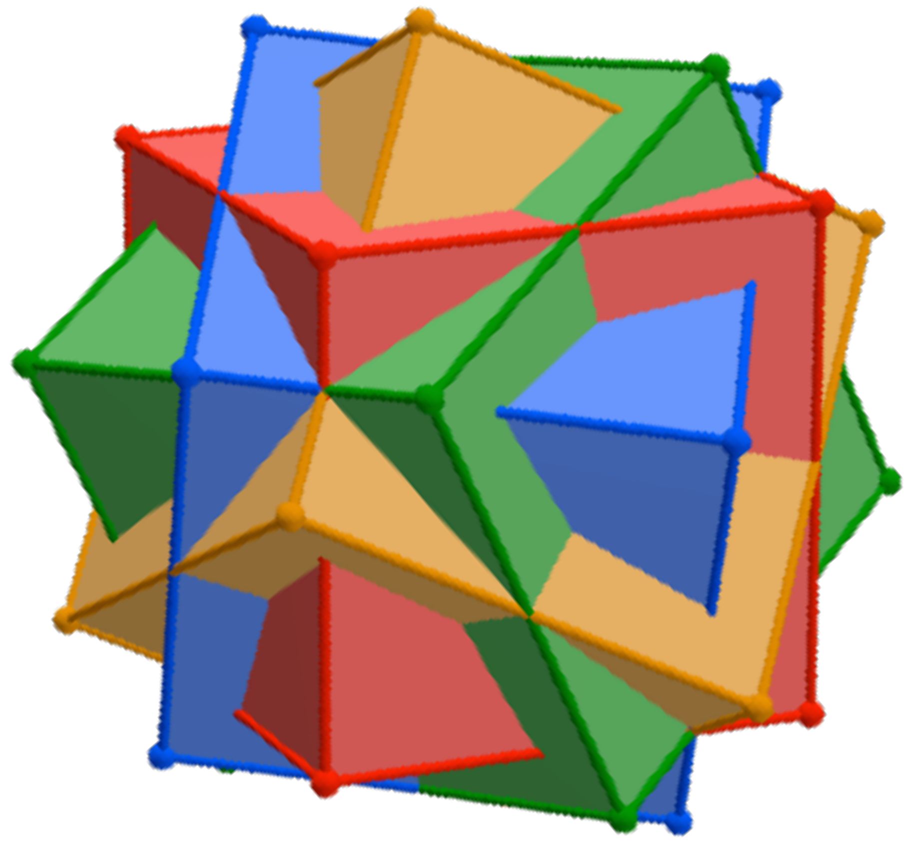
How to rotate a vector $\mathbf{v} = (x, y, z) \in \mathbb{R}^3$?

- Interpret as a purely imaginary quaternion $\mathbf{v} = 0 + x\mathbf{i} + y\mathbf{j} + z\mathbf{k}$.
- Then the rotated vector is $\mathbf{qvq}^{-1}$ (= $\mathbf{qvq}^*$ because $|\mathbf{q}| = 1$). Surprisingly, still purely imaginary!

Actually $\mathbf{q}$ and $-\mathbf{q}$ represent the same rotation…



Keenan Crane

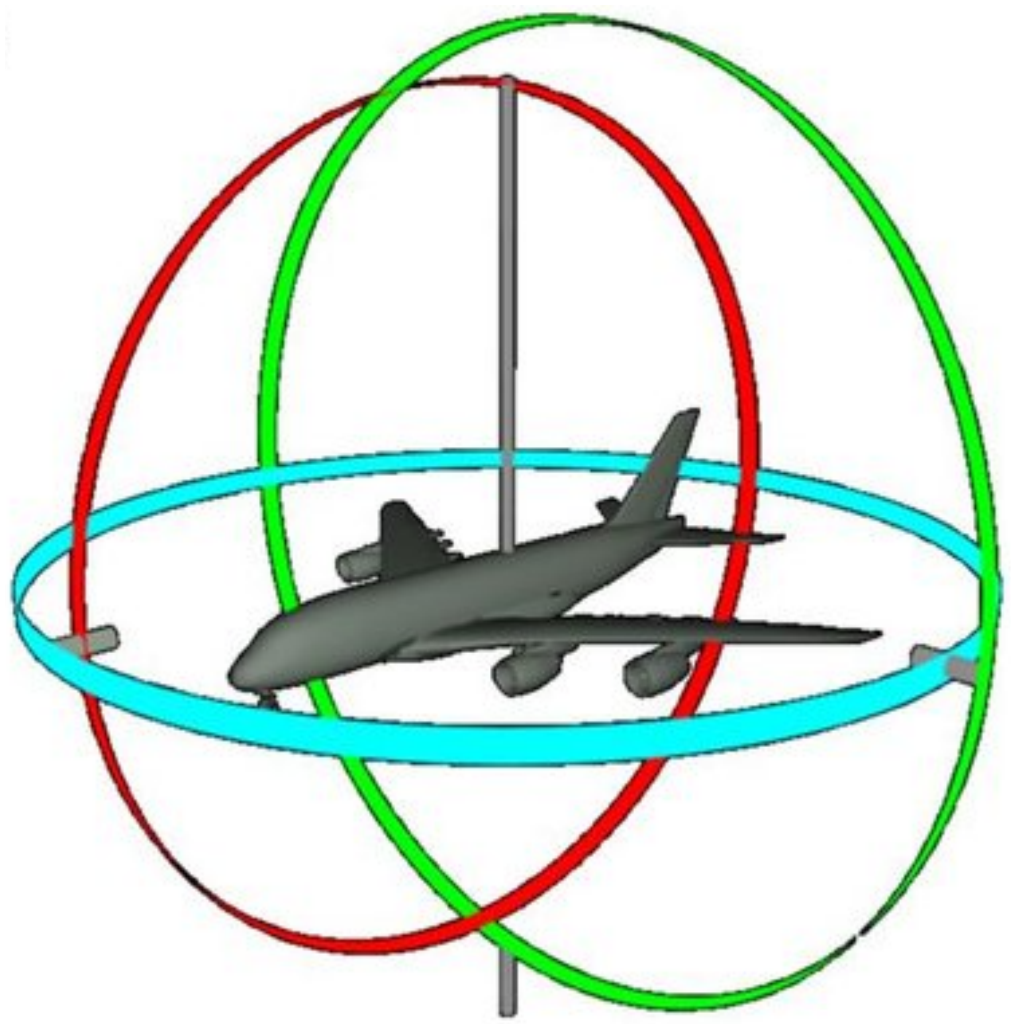


Puzzle:

In complex numbers we had $z_1 z_2 = z_2 z_1$, but in quaternions we generally have $\mathbf{q}_1 \mathbf{q}_2 \neq \mathbf{q}_2 \mathbf{q}_1$.

Why is this a very good thing, if we intend to use quaternions to model 3D rotations?

…And why did that reason not apply in 2D?



No gimbal lock (unlike Euler angles)

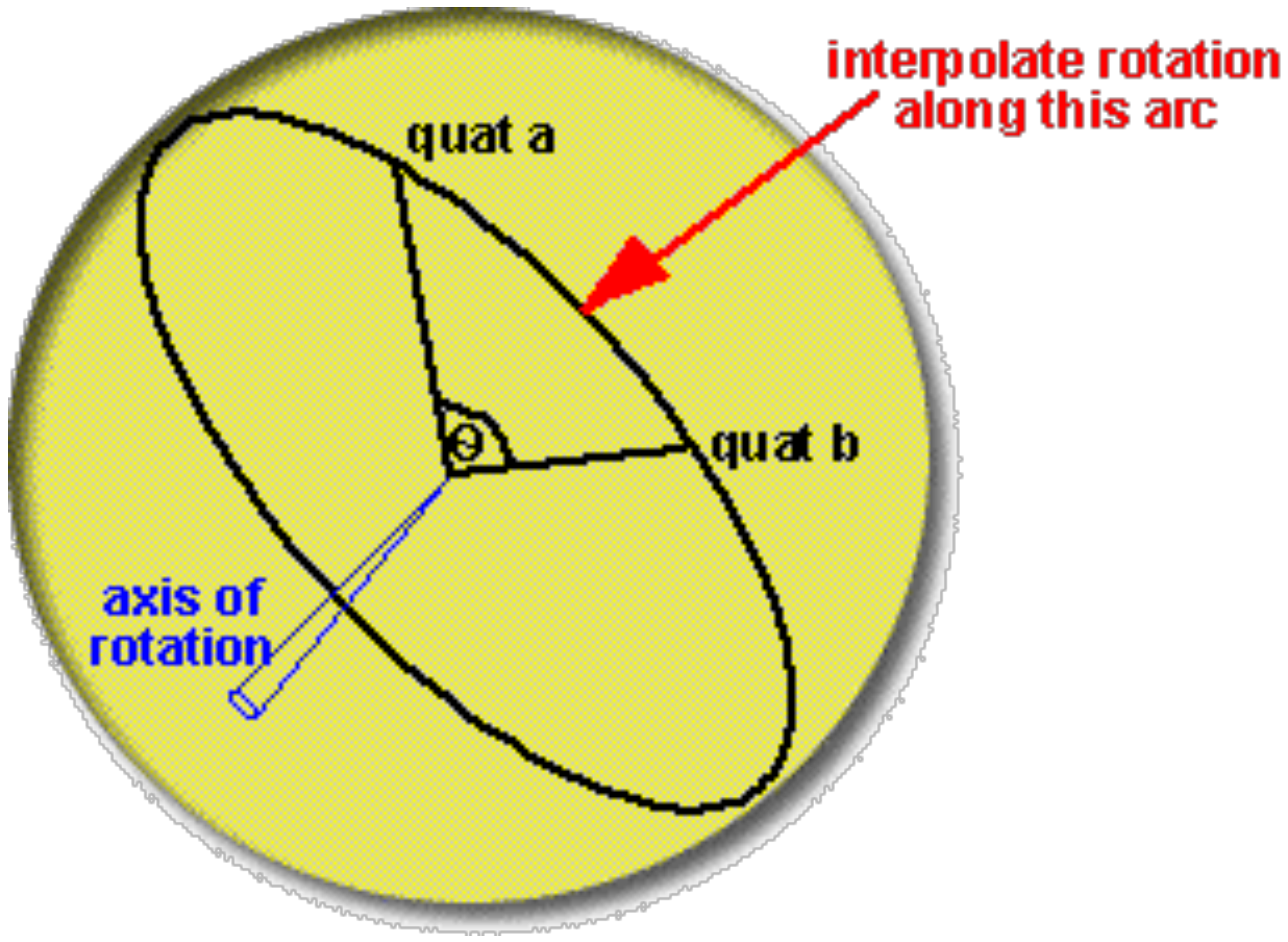
Easy to normalize (unlike rotation matrices)

Easy to interpolate via spherical linear interpolation ("**Slerp**"):

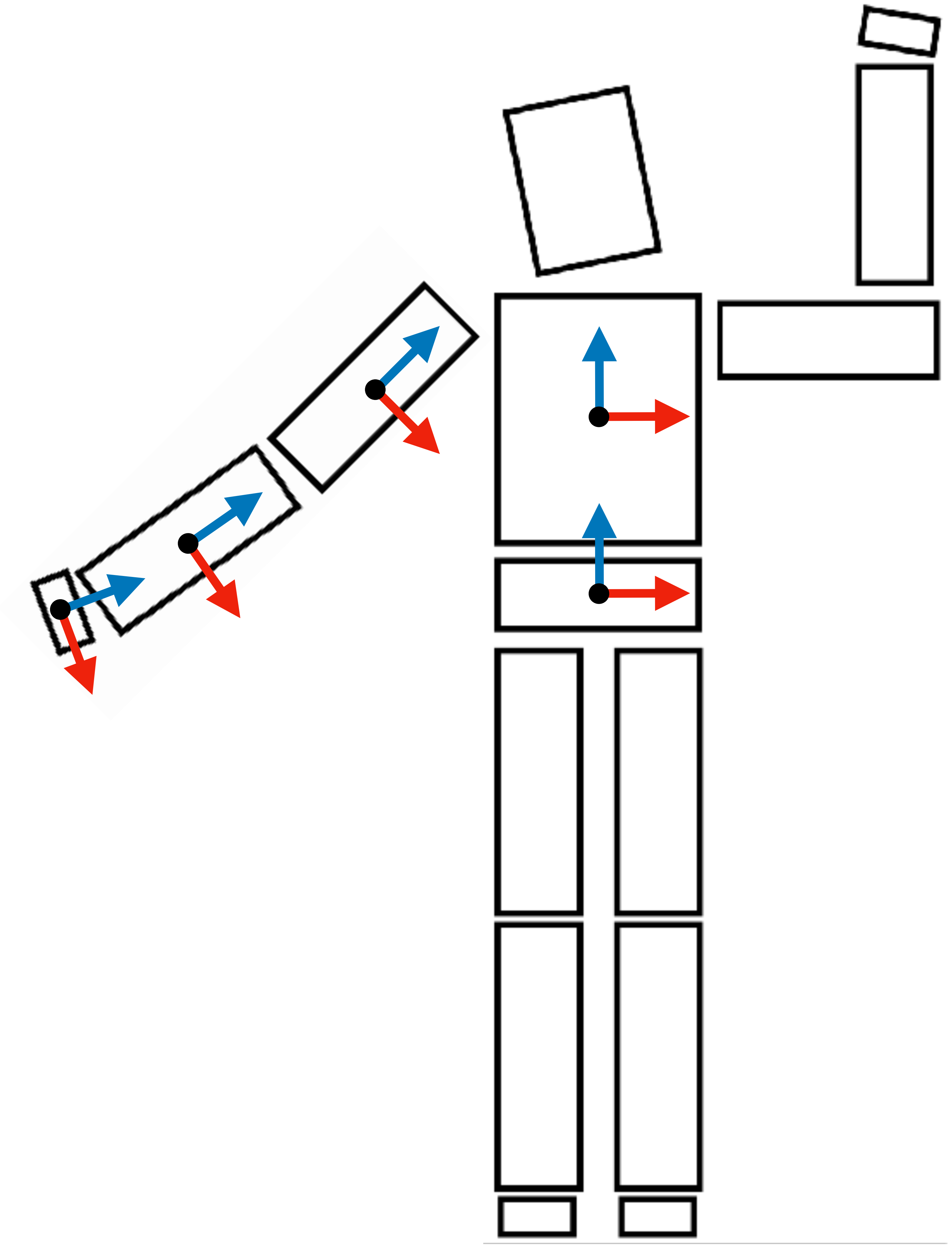
$$\mathrm{Slerp}(\mathbf{q}_0, \mathbf{q}_1, t) = (\mathbf{q}_1\, \mathbf{q}_0^{-1})^t\, \mathbf{q}_0$$

$$= \frac{\sin((1-t)\Omega)}{\sin\Omega}\mathbf{q}_0 + \frac{\sin(t\Omega)}{\sin\Omega}\mathbf{q}_1$$

where $\cos\Omega = \mathbf{q}_0 \cdot \mathbf{q}_1$ treated as vectors



Generalized coordinates

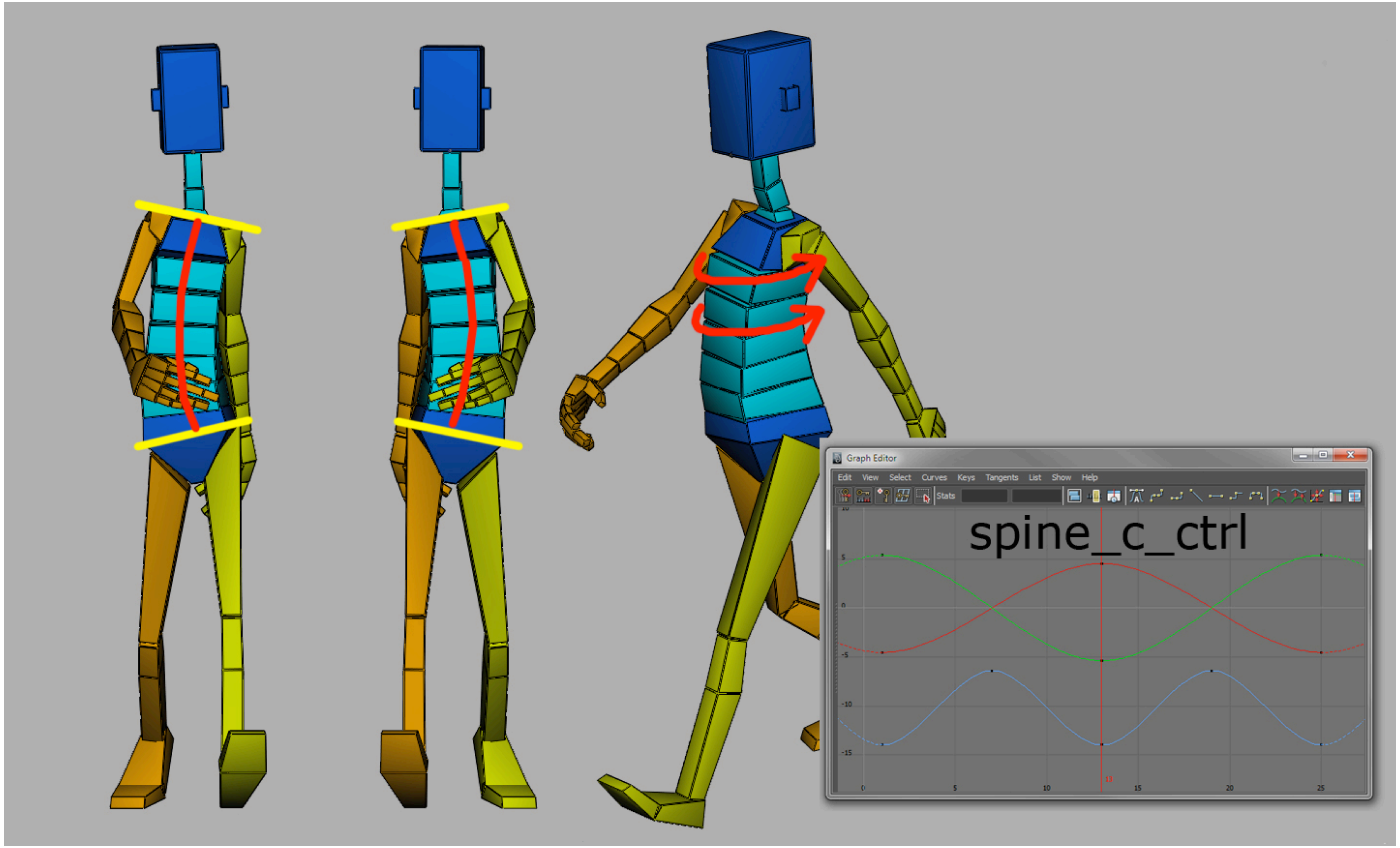


A single vector to specify the pose of the entire body

$$\mathbf{q} = (q_1, q_2, q_3, q_4, \ldots, q_n)$$

- Location of root
- Orientation of root
- Joint angles of all joints

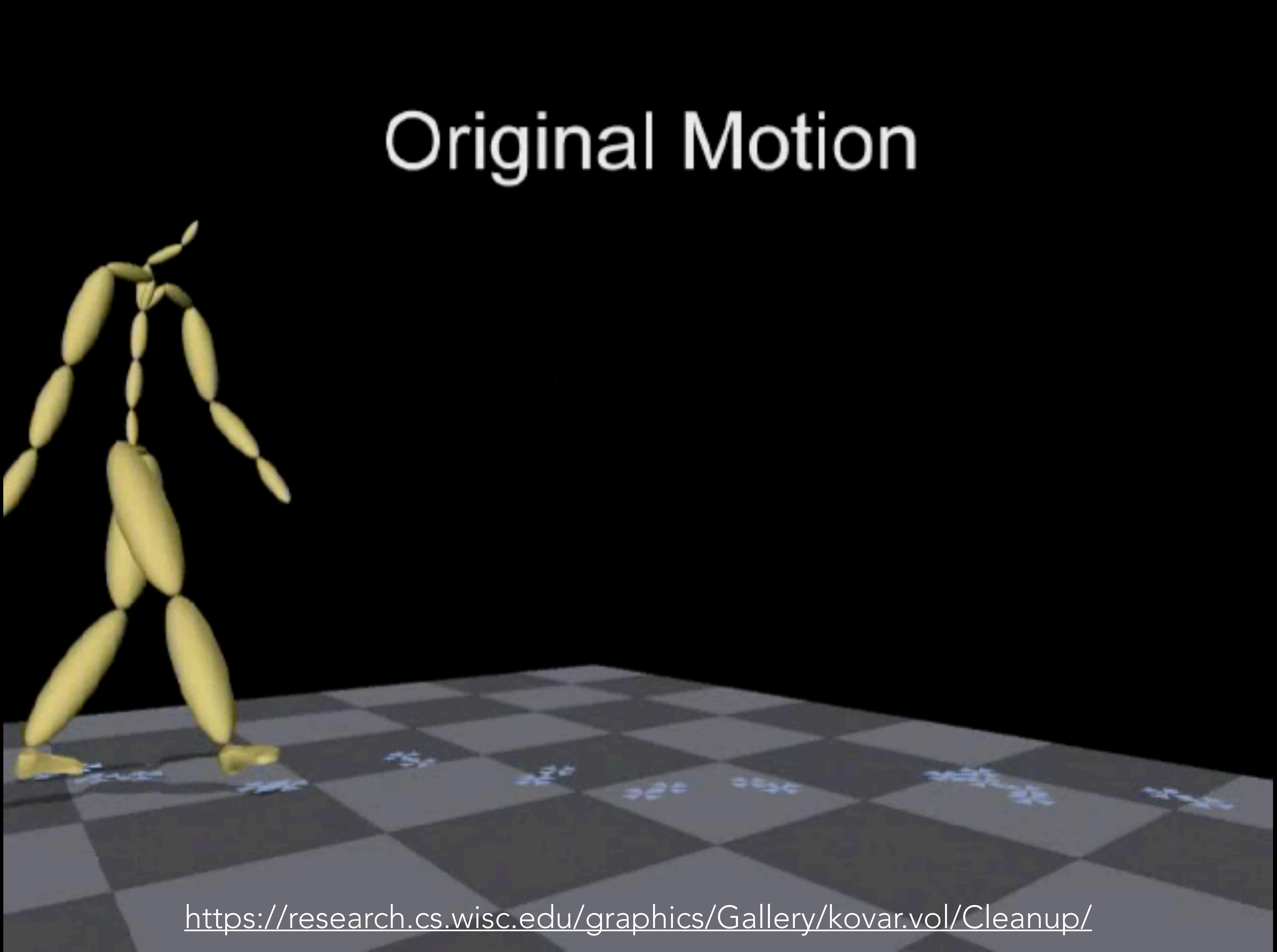
To animate a character, specify function $t \mapsto \mathbf{q}$


Graph Editor
Edit View Select Curves Keys Tangents List Show Help
Stats
spine_c_ctrl

Given joint angles, compute transformation of points: **forward kinematics**

Liu 2009

Given desired transformation of end points, how to find the joint angles that achieve it?
Inverse kinematics



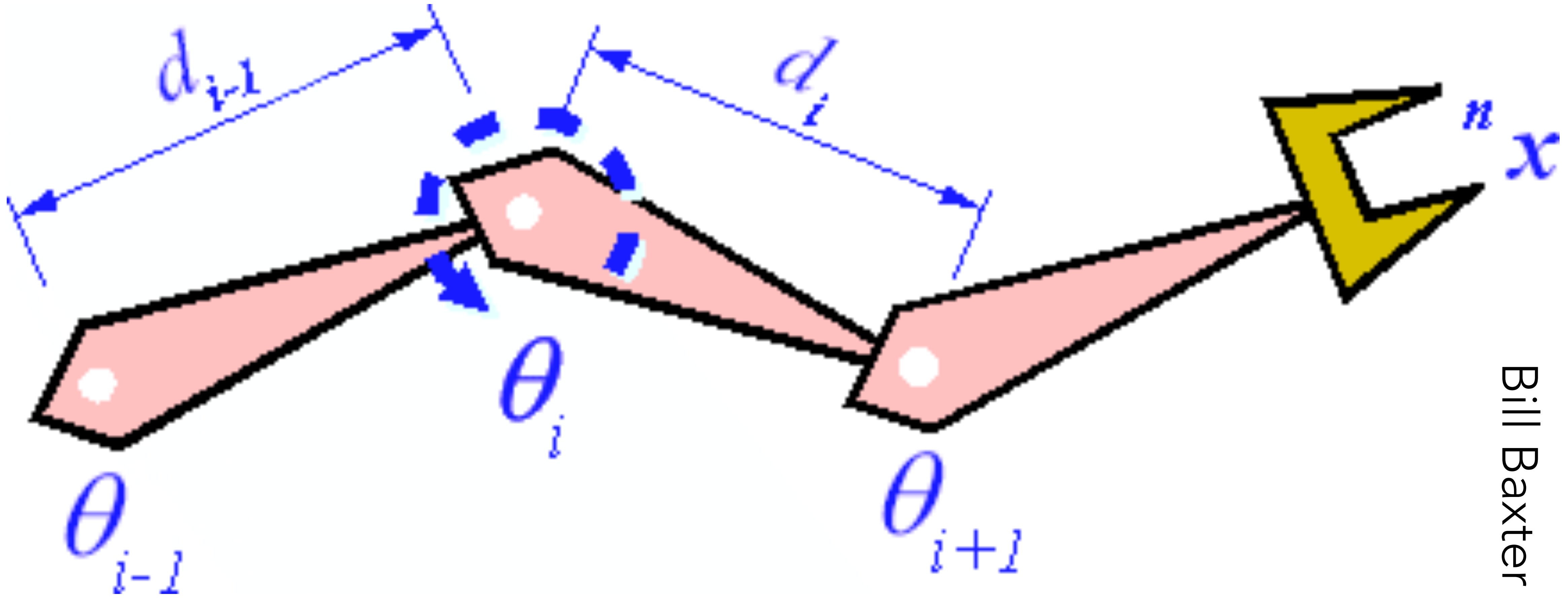
https://research.cs.wisc.edu/graphics/Gallery/kovar.vol/Cleanup/

Kovar et al. 2002

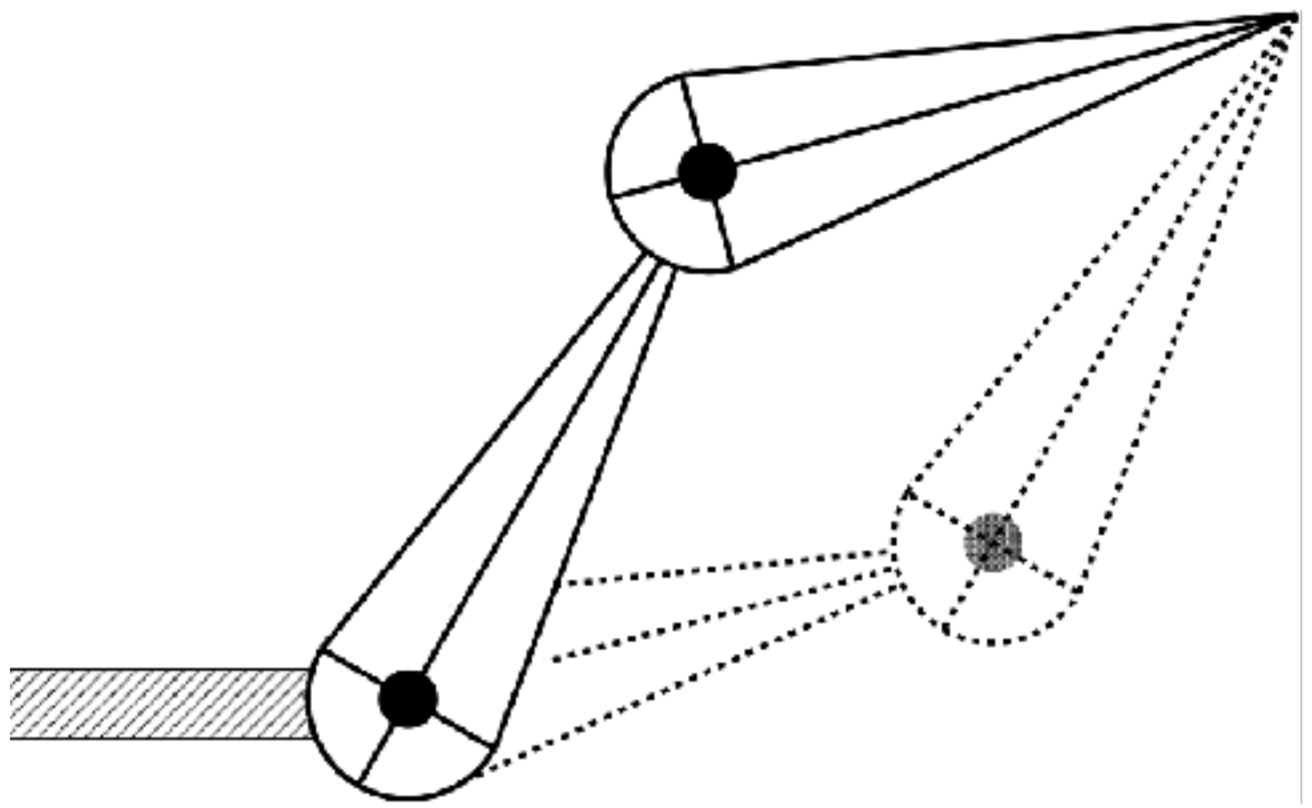
Position of target point (**end effector**) depends nonlinearly on joint angles

$\mathbf{x} = f(\mathbf{q})$

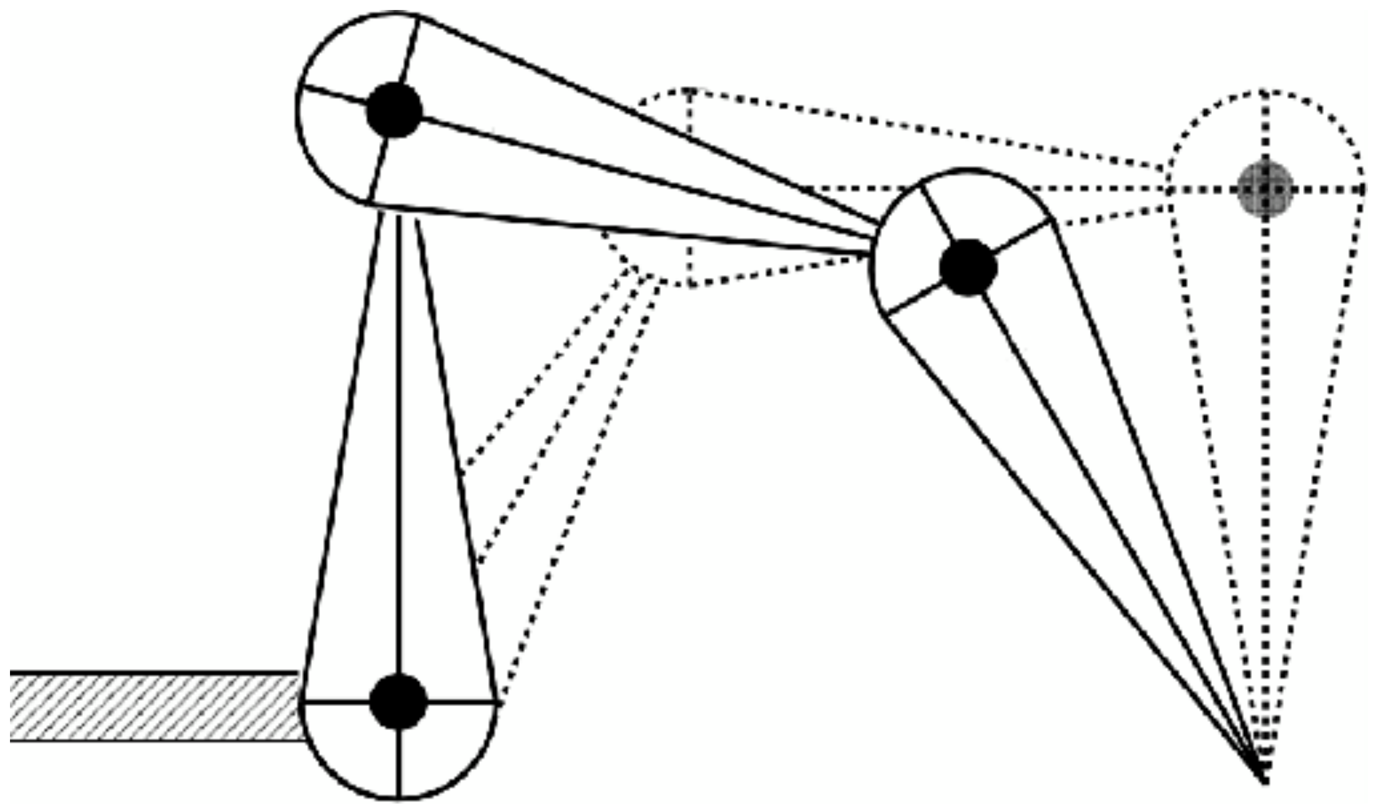
Given desired $\mathbf{x}$, how to compute $\mathbf{q}$?



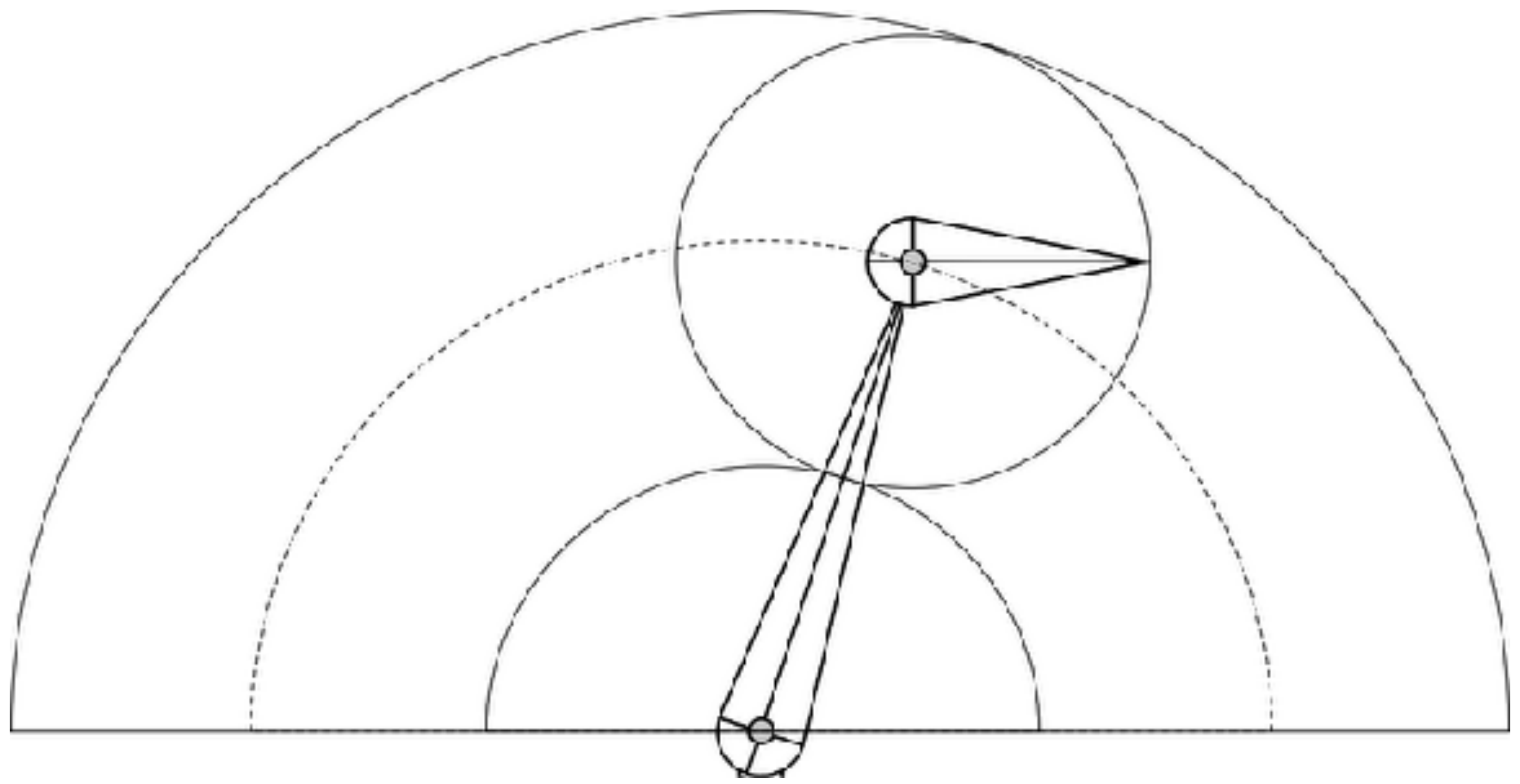
Bill Baxter



Multiple solutions



Infinitely many solutions



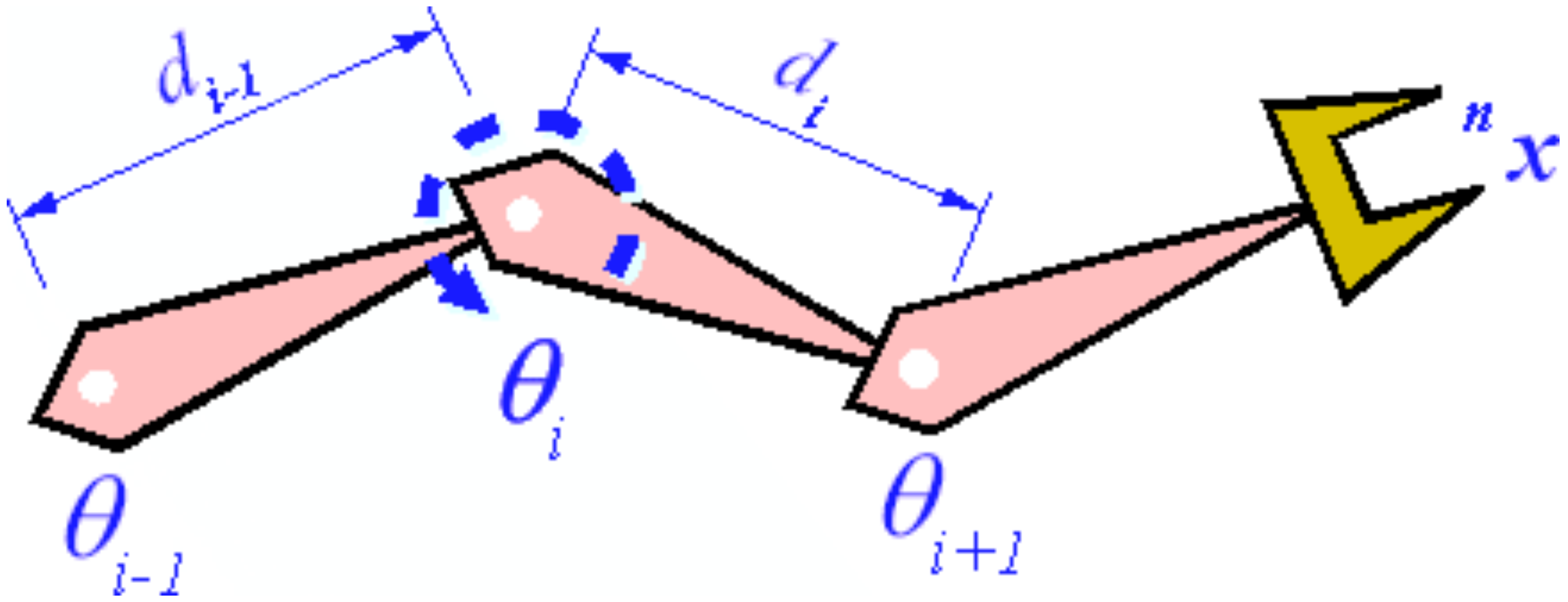
No solution

James O'Brien

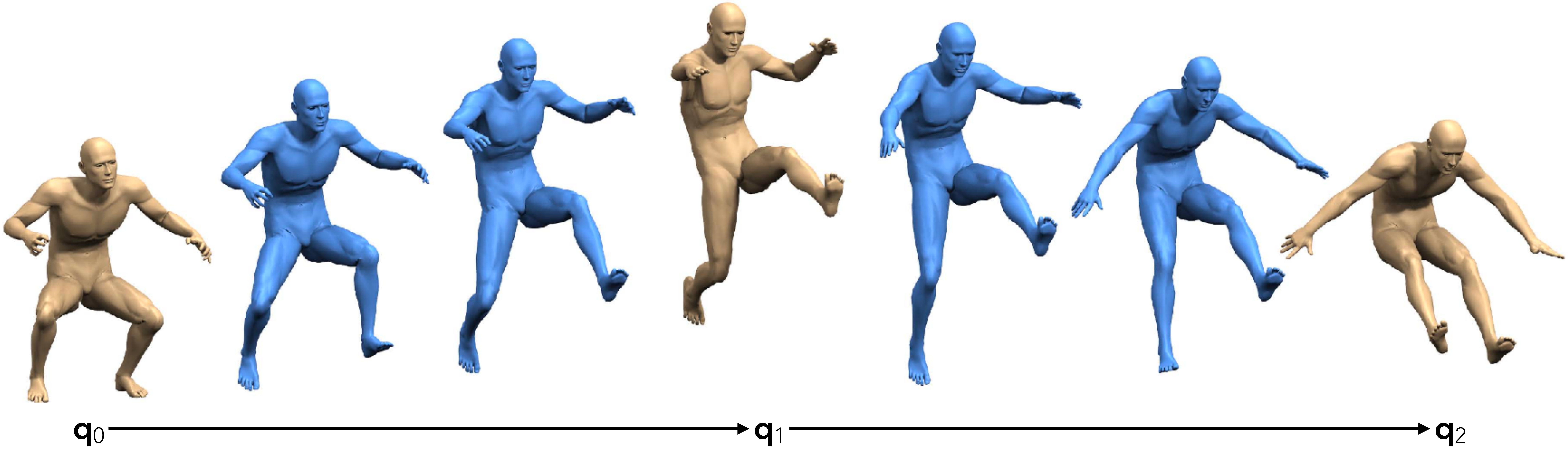
Analytical solutions only in special cases (1 or 2 bones)!

Use your favourite numerical method to solve $f(\mathbf{q}) = \mathbf{x}$ or to minimize $\|f(\mathbf{q}) - \mathbf{x}\|^2$

- Cyclic coordinate descent
- Pseudoinverse of Jacobian $\mathbf{J} = \left[\frac{\partial x_i}{\partial q_j}\right]$
- Damped least squares
- …



Keyframe animation



Chu & Lee 2009

How to interpolate the animation controls / generalized coordinates over time?