Note 12-Animation

Metaverse

Interpolation based animation

3 Types of Equations

Explicit y = f(x)

good for generating points (y values from x values)

dependent on the choice of the coordinate axis

sometimes ambiguous, e.g. $y = \sqrt{x}$

Implicit f(x,y) = 0

good for testing to see if a point is on a curve

by testing f(x,y) = 0 for a given point (x, y)

not generative (inconvenient to render the curve)

Parametric c(t) = (x(t), y(t))

good for rendering

“on” testing is not easy

Polynomial Equations

Contain only variables raised to a power

ex) $x^5 + 3x^4 – 4 x^2 + 1$

Linear Equation

Highest power is 1, ex) $2x + 3$, $-5x$, …

Quadratic Equation

Highest power is 2, ex) $3x^2 + 4x – 1$, $-x^2$, $6x^2 + 5$, …

Cubic Equation

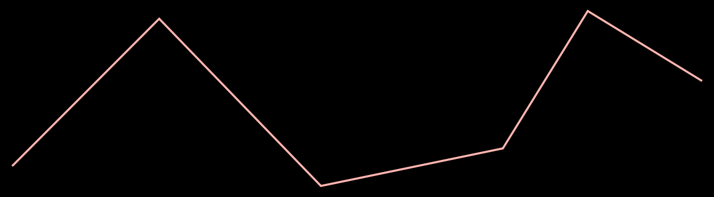
Highest power is 3

Continuity

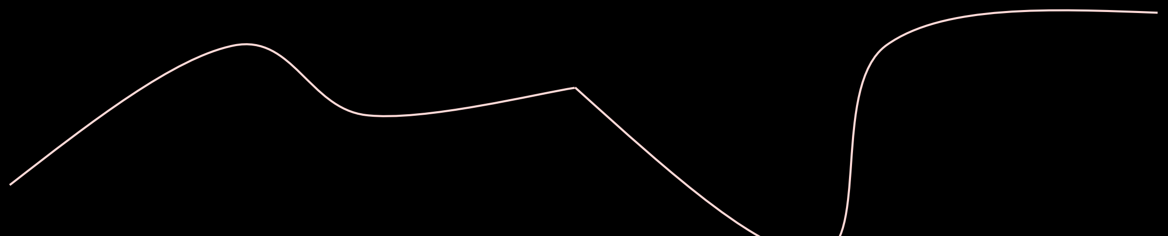
How well behaved the curve is in a mathematical sense.

C^0 continuous: positional continuity

ex) F(t) is C^0 continuous for all t. (piecewise linear)



ex) F(t) and G(t) is C^0 continuous at t_0



Continuity

C^1 continuous: tangential continuity

ex) $F(t)$ and $G(t)$ are C^1 continuous at t_0

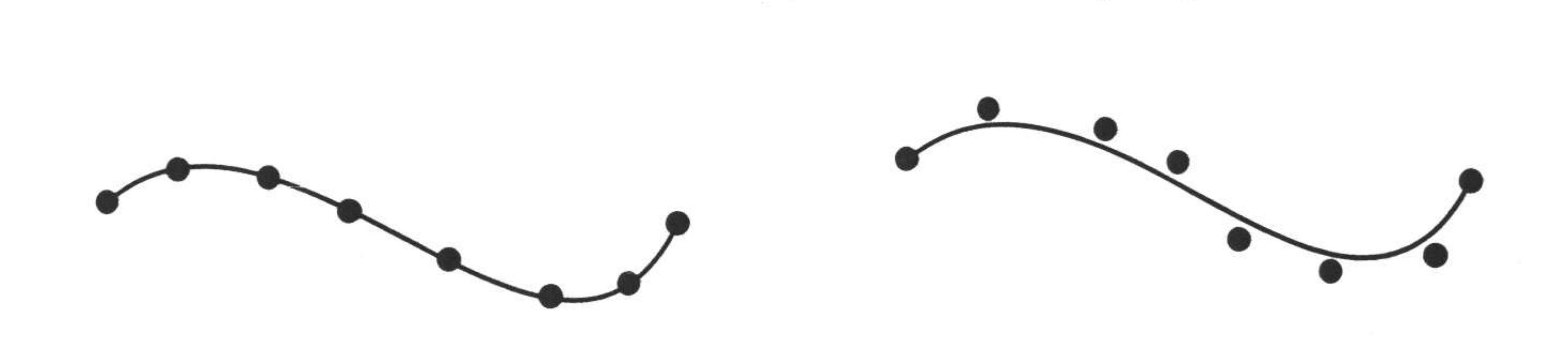
$F'(t_0) = G'(t_0)$

C^2 continuous: curvature continuity

ex) $F(t)$ and $G(t)$ are C^2 continuous at t_0

$F''(t_0) = G''(t_0)$

Interpolation vs. Approximation



Interpolation

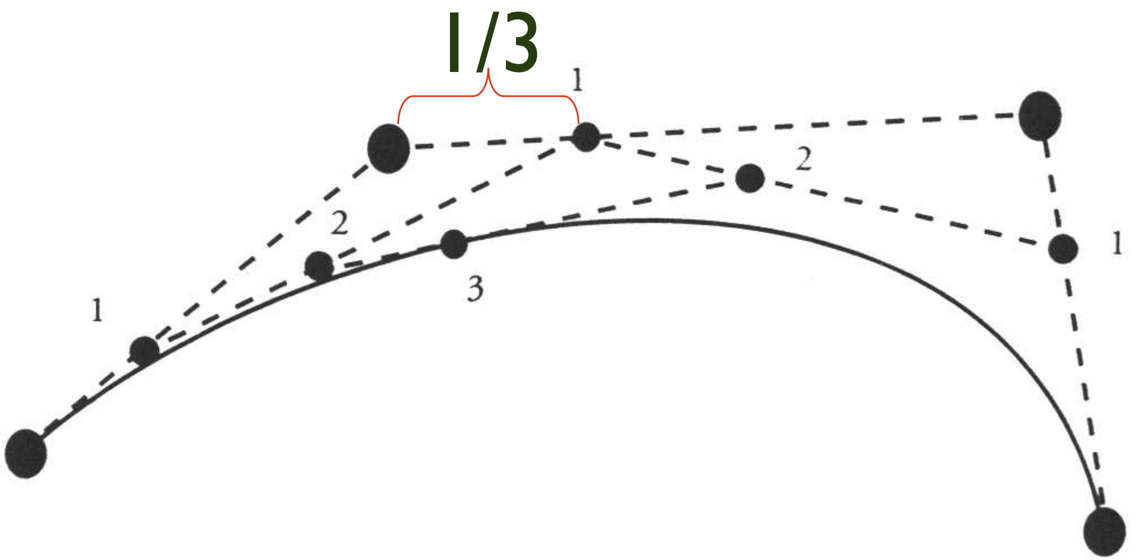
Approximation

De Casteljau Construction

Bezier curve construction

Example

How to compute P(1/3) ?



Interpolation steps

1. 1/3 of the way between paired points
2. 1/3 of the way between points of step 1
3. 1/3 of the way between points of step 2

Figure B.36 De Casteljau construction of a point on a cubic Bezier curve

Composite Bezier Curve

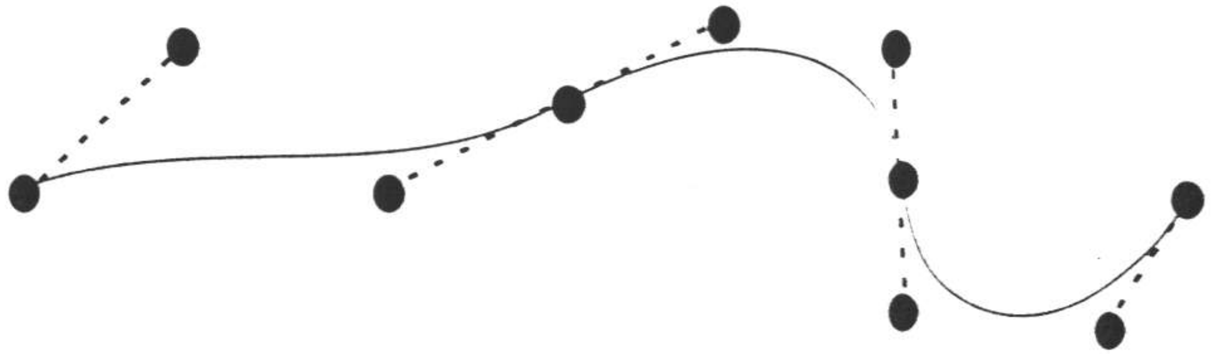
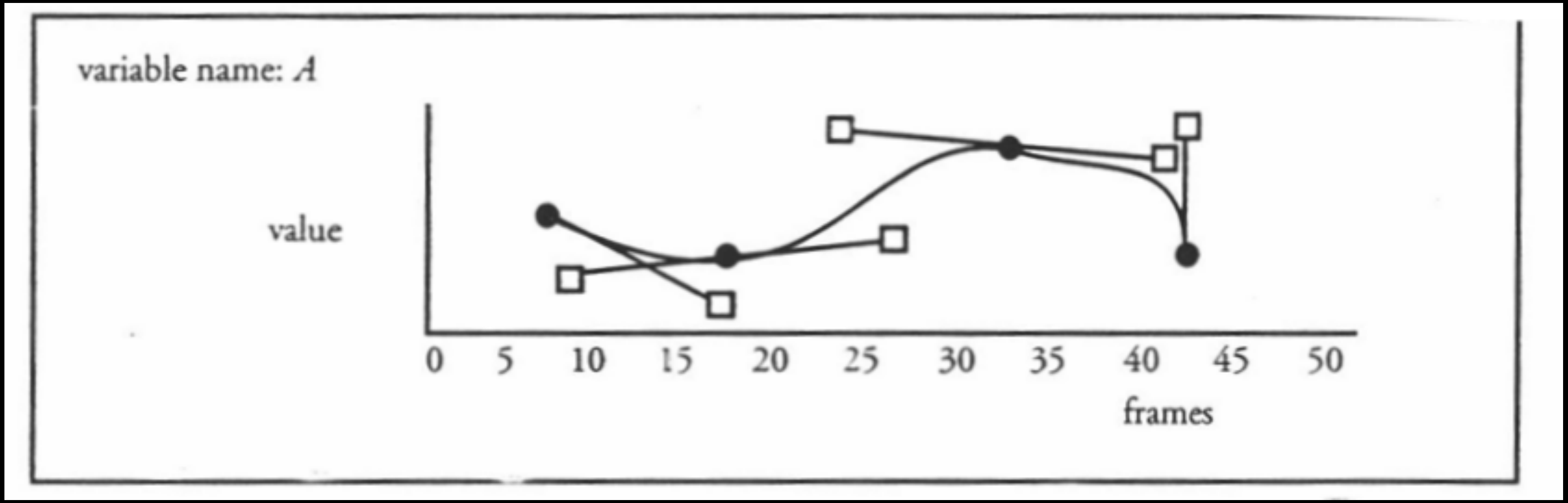
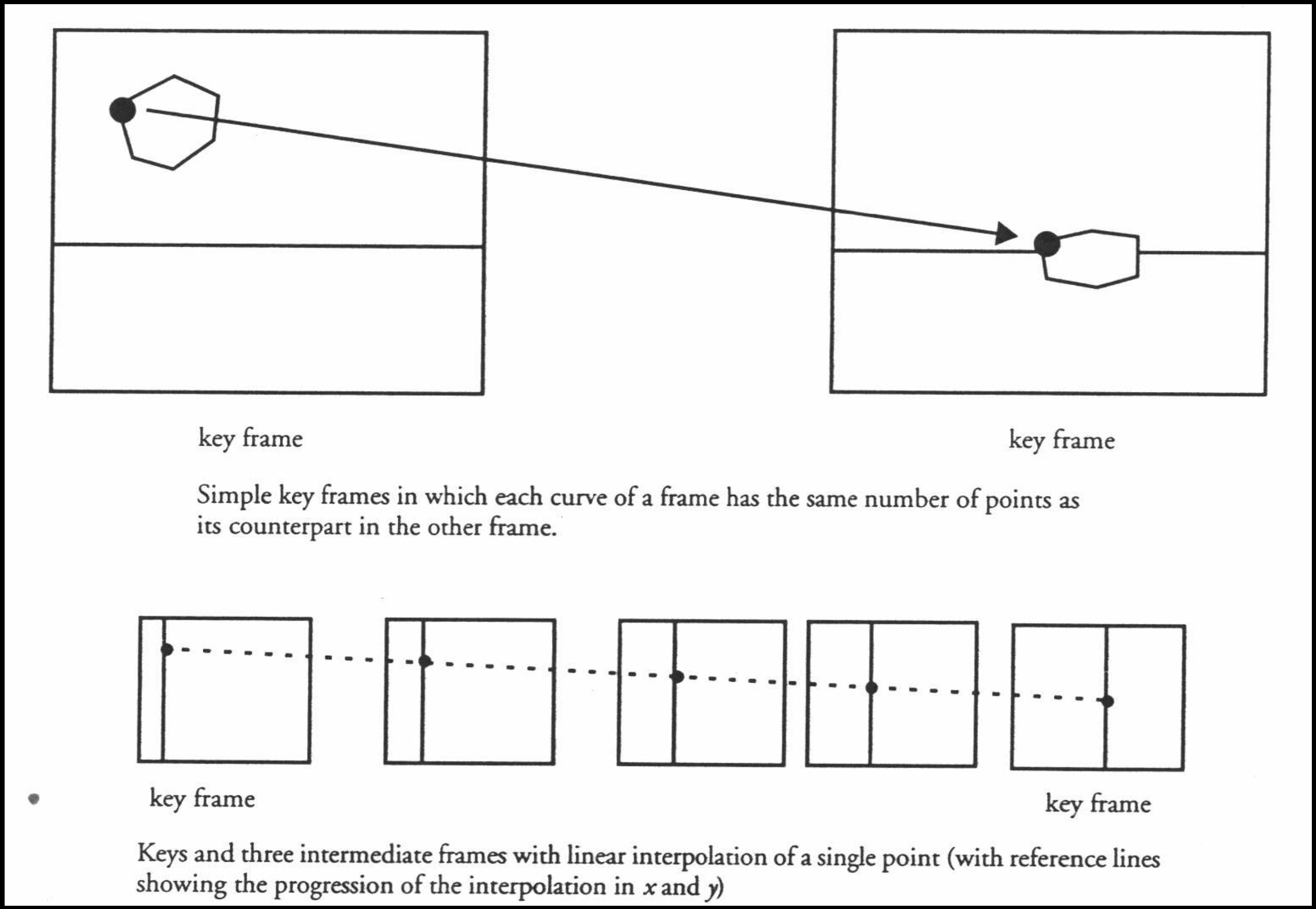


Figure B.35 Composite cubic Bezier curve showing tangents and colinear control points

Key-Frame Systems



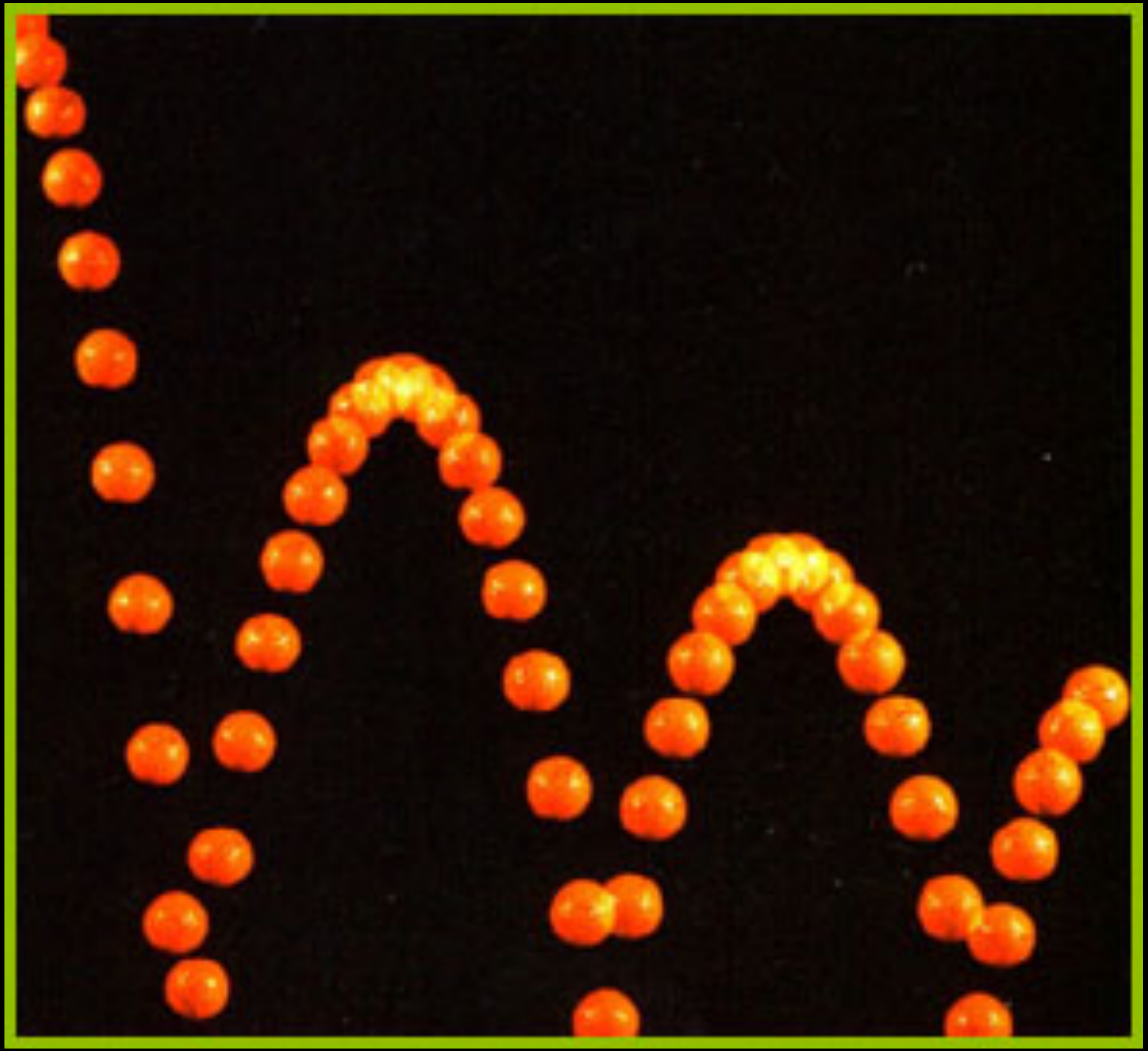
Interpolating Key Frame Shapes – Linear Interpolation



Simple key frames in which each curve of a frame has the same number of points as its counterpart in the other frame.

Keys and three intermediate frames with linear interpolation of a single point (with reference lines showing the progression of the interpolation in *x* and *y*)

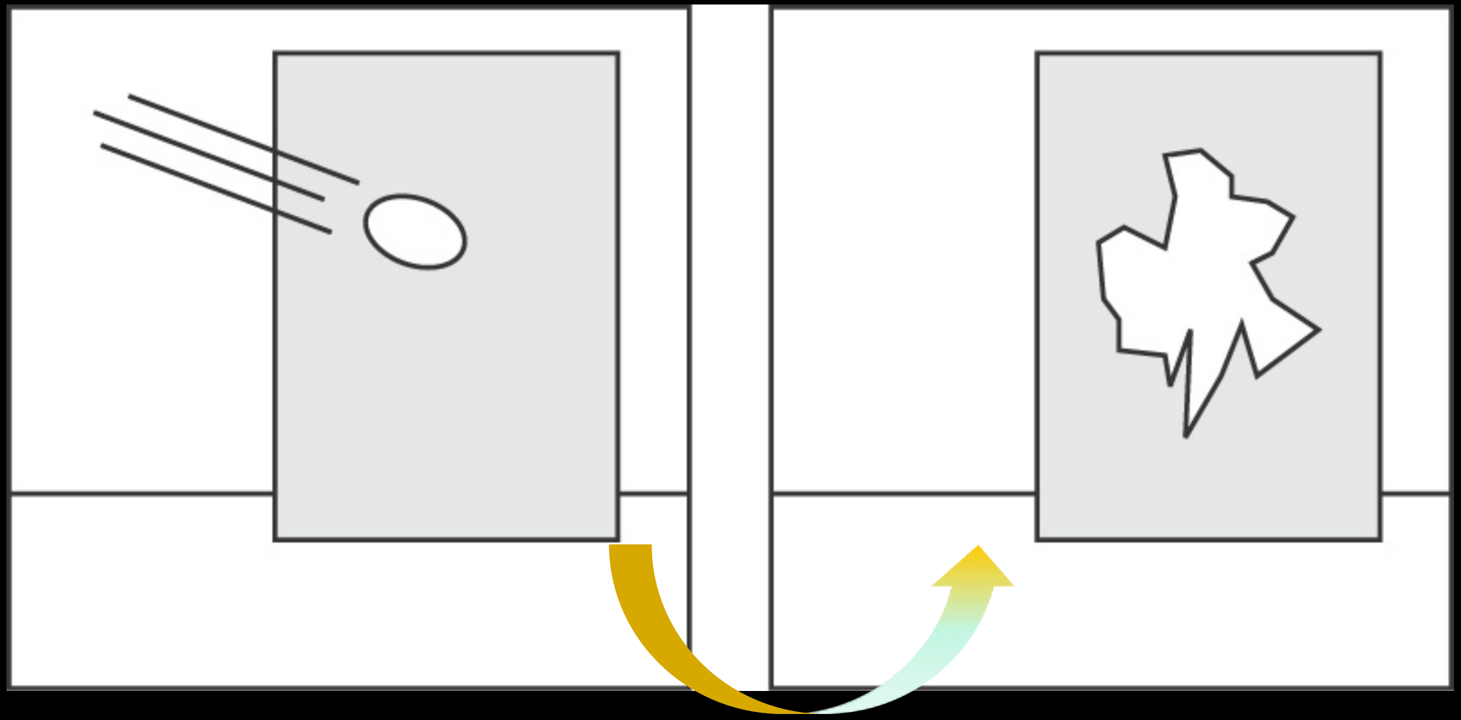
Simple animation



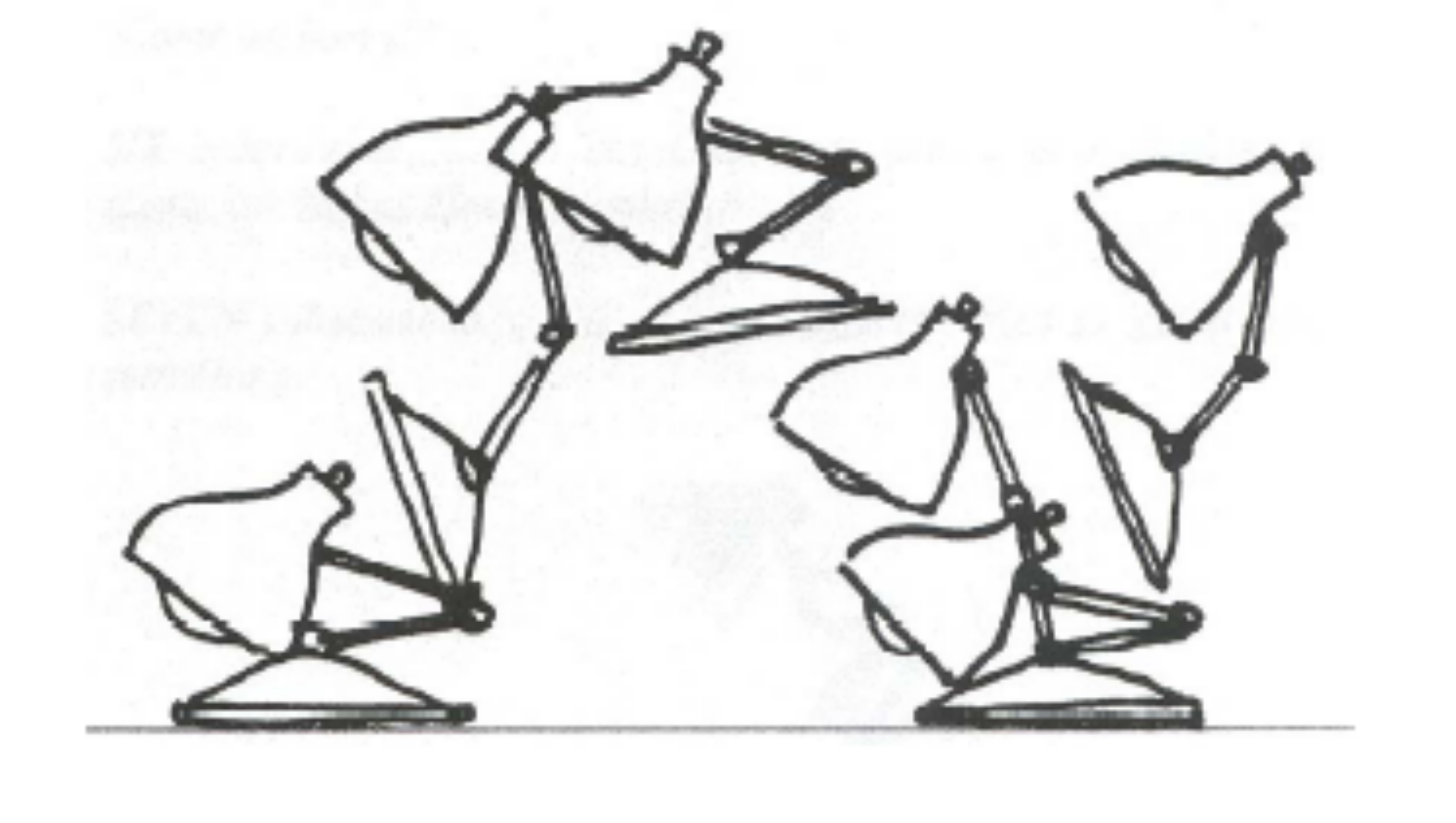
Complex Shape Interpolation

We don’t know which vertex should go to which vertex.

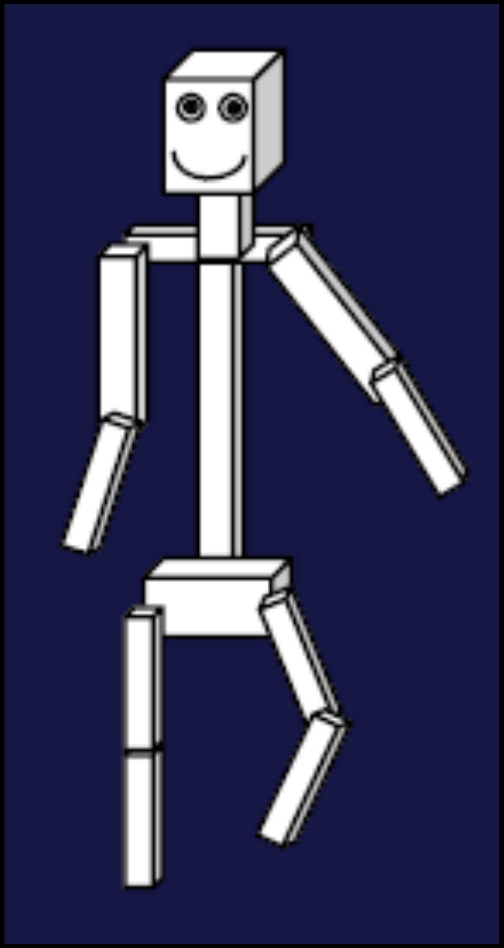
Usually polygonal shape interpolation is not easy.

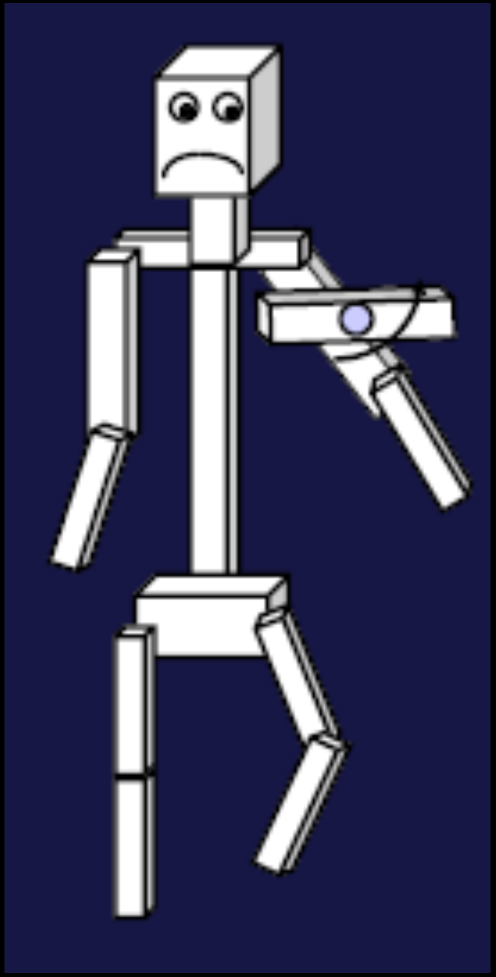


??



Kinematics of articulated figures



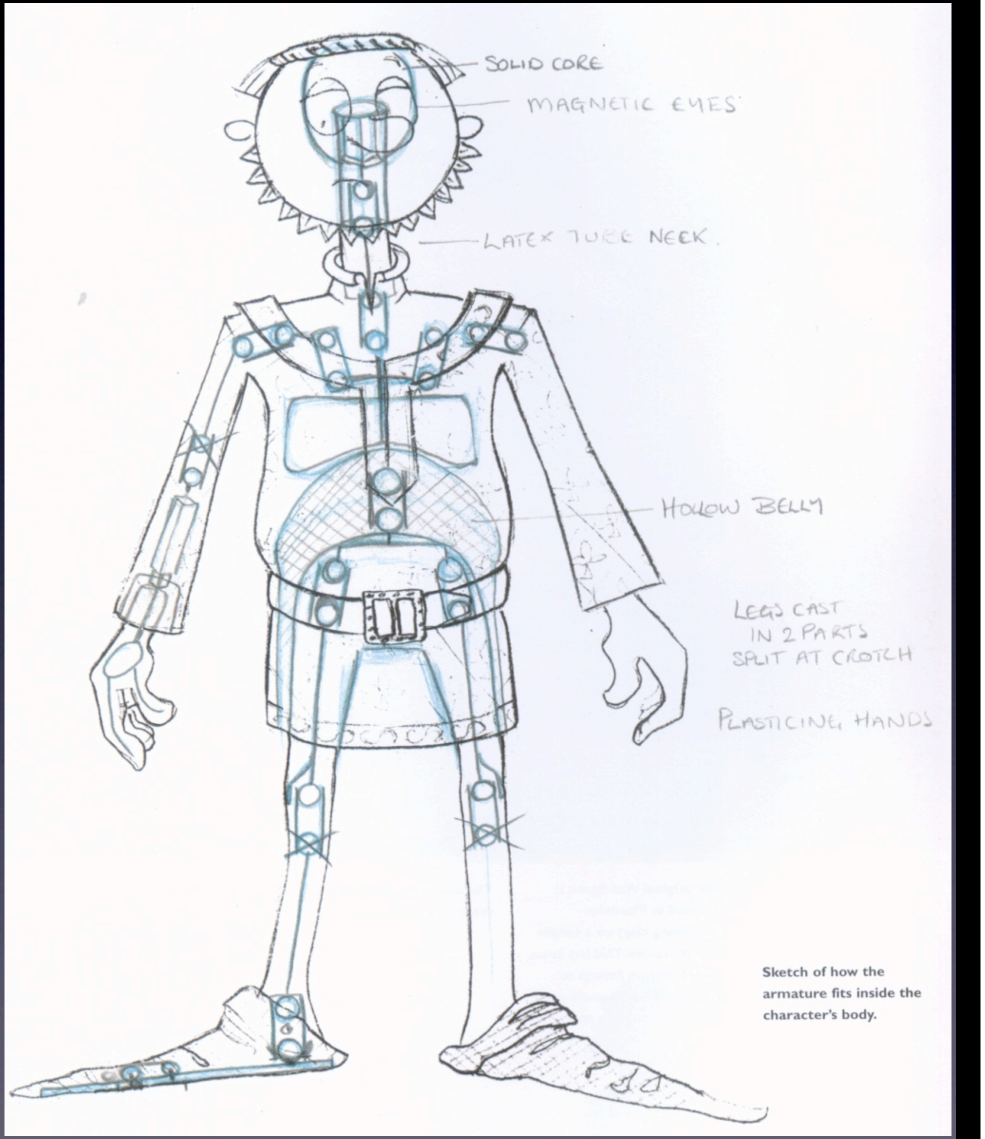


Kinematics

- How to animate skeletons (articulated figures)

- ***Kinematics*** is the study of motion without regard to the forces that caused it

- <--> dynamics



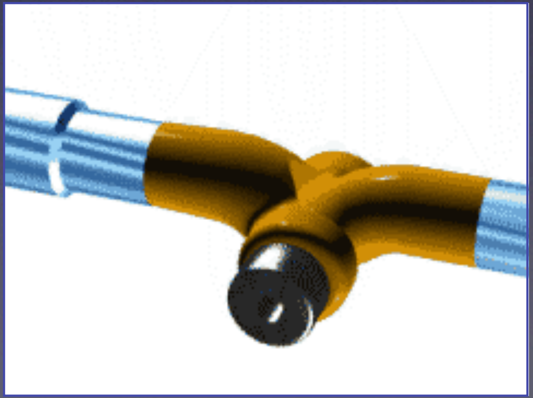
Sketch of how the armature fits inside the character's body.

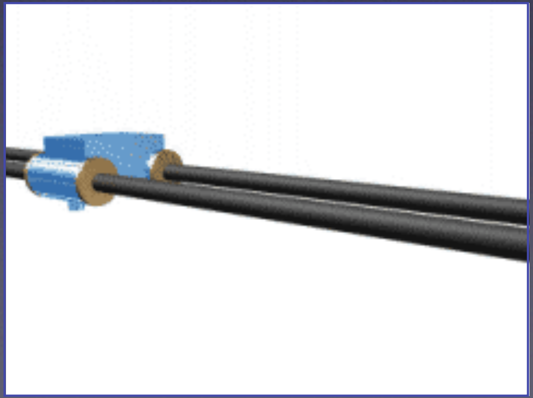
Hierarchical Models

Tree structure of joints and links

- The root link can be chosen arbitrarily

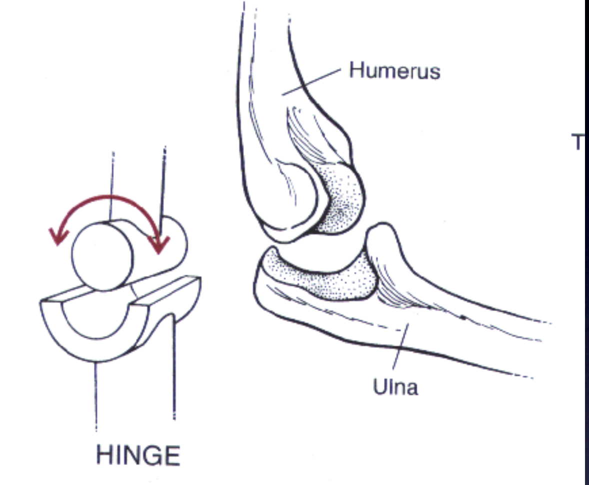
- Joints
  - Revolute (hinge) joint allows rotation about a fixed axis
  - Prismatic joint allows translation along a line
  - Ball-and-socket joint allows rotation about an arbitrary axis

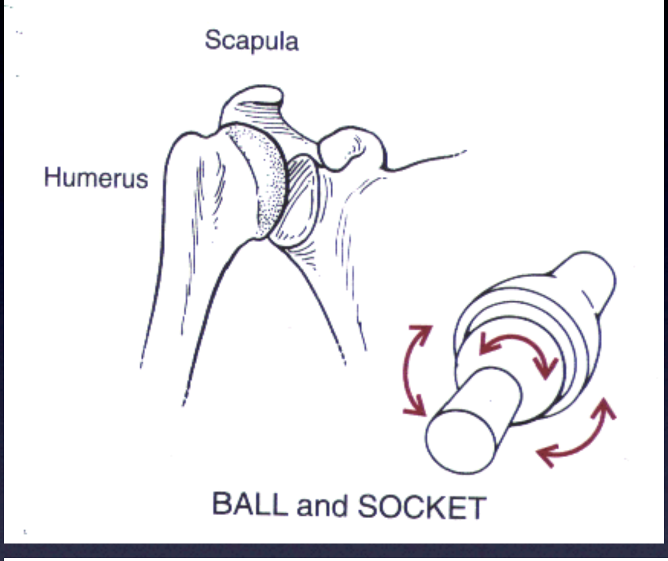


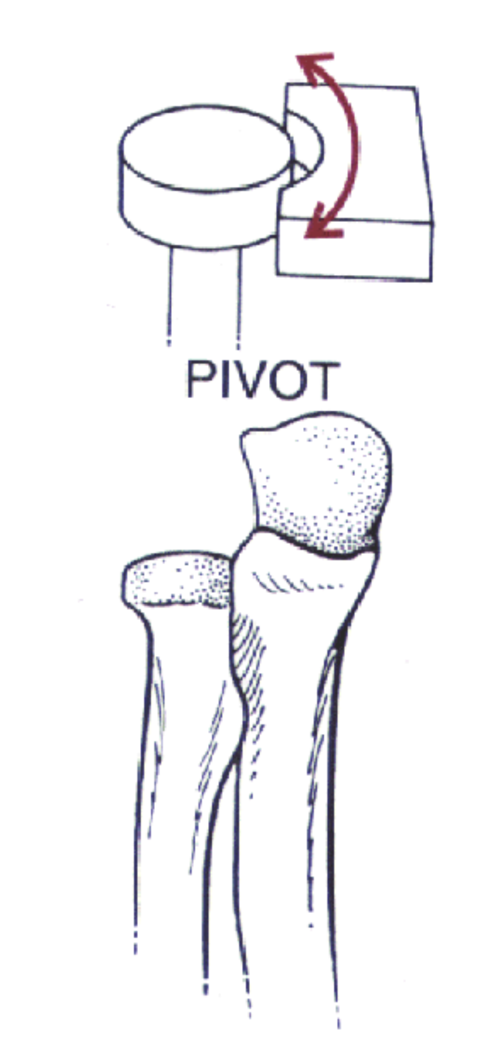


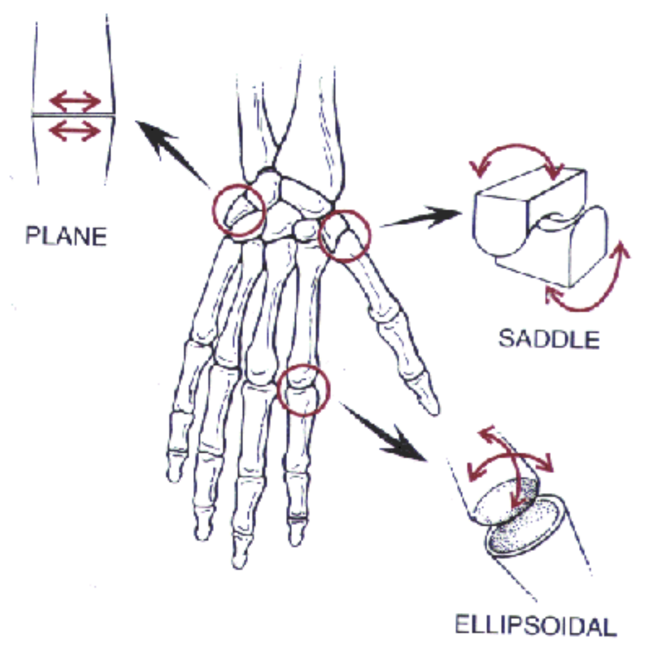
Human Joints

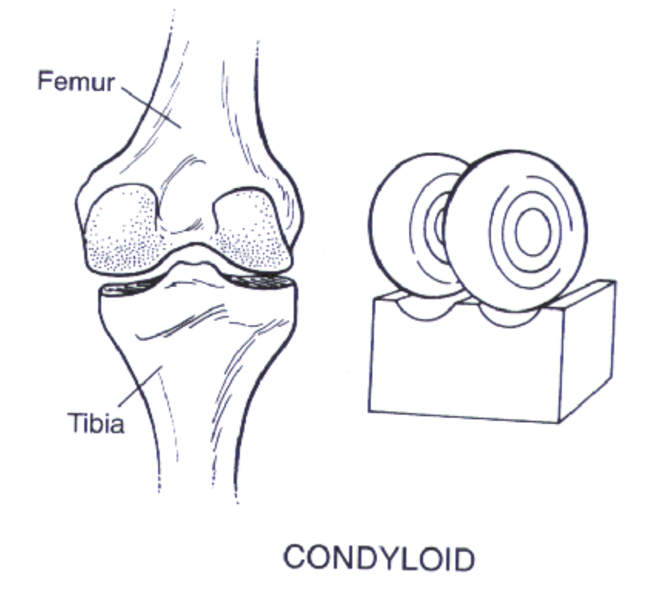
Human joints are actually much more complicated



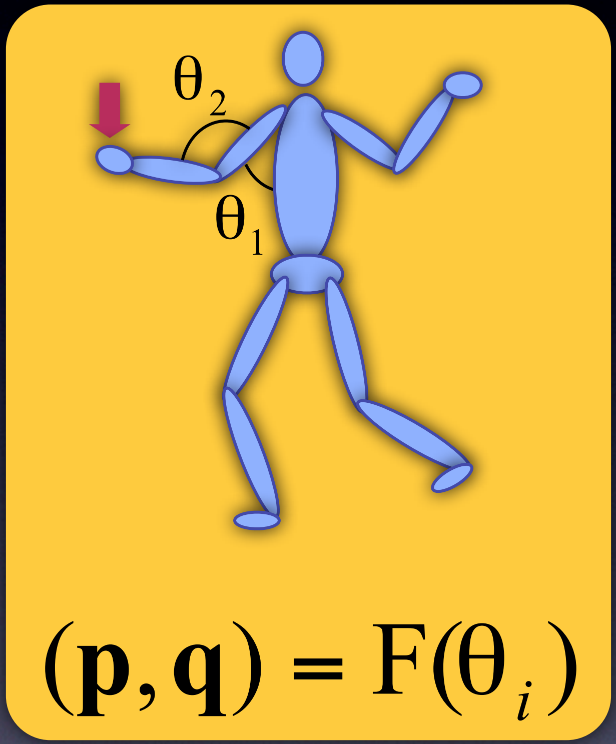






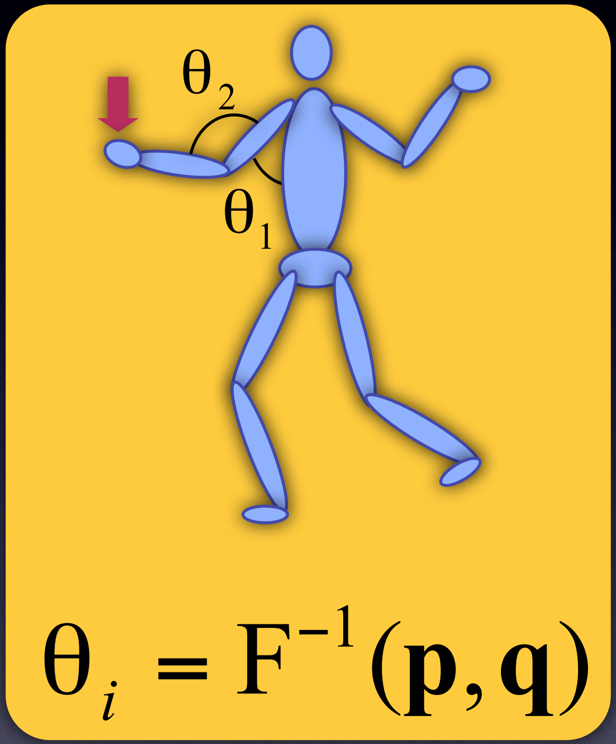


Forward and Inverse Kinematics



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

Forward Kinematics



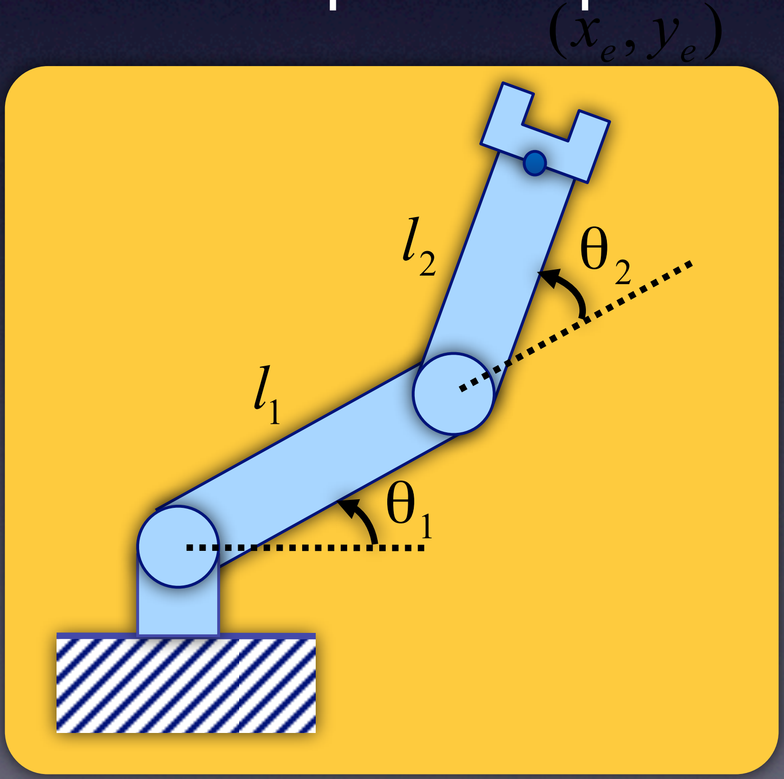
$$\theta_i = F^{-1}(\mathbf{p},\mathbf{q})$$

Inverse Kinematics

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

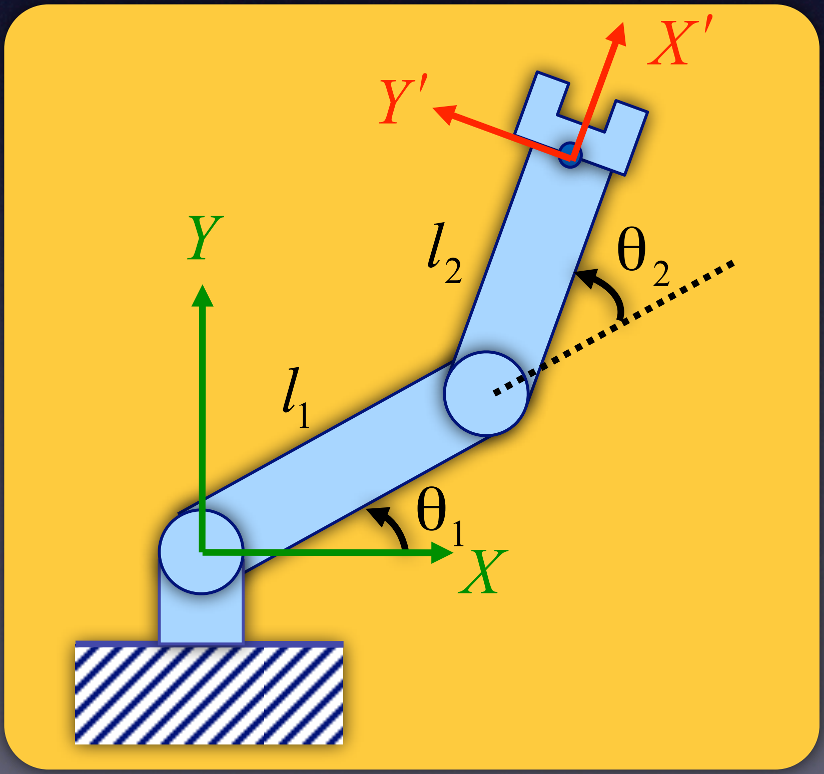


$$x_e = l_1 \cos\theta_1 + l_2 \cos(\theta_1 + \theta_2)$$
$$y_e = l_1 \sin\theta_1 + l_2 \sin(\theta_1 + \theta_2)$$

Forward Kinematics: A Simple Example

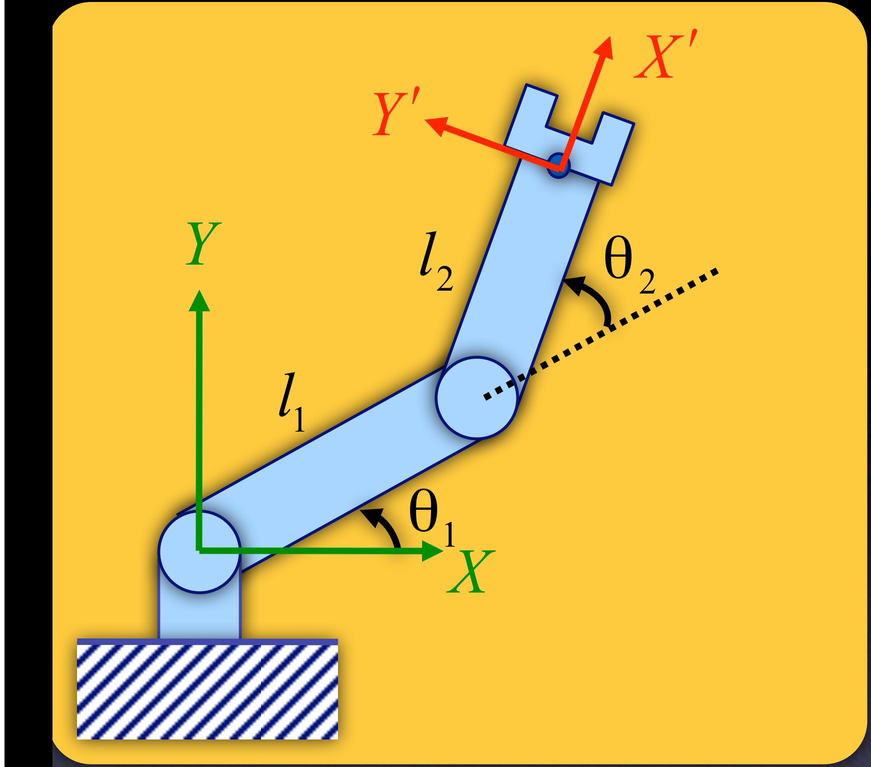
Forward kinematics map as a coordinate transformation

- Forward kinematics map transforms the position and orientation of the end-effector according to joint angles



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

A Chain of Transformations



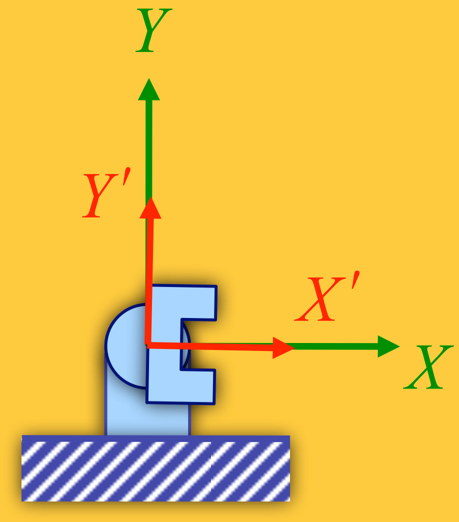
$$\begin{pmatrix} x_e \\ y_e \\ 1 \end{pmatrix} = \begin{pmatrix} & & \\ & T & \\ & & \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$$

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

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

Thinking of Transformations

In a view of global coordinate system

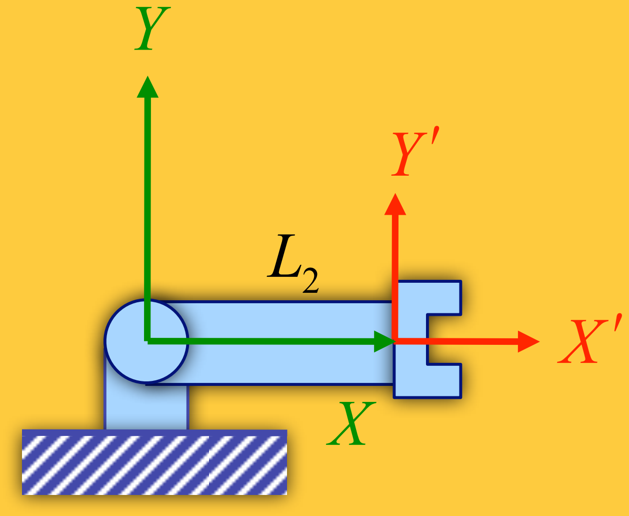


$$T = (rot\theta_1)(transl_1)(rot\theta_2)(transl_2)$$

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

Thinking of Transformations

- In a view of global coordinate system

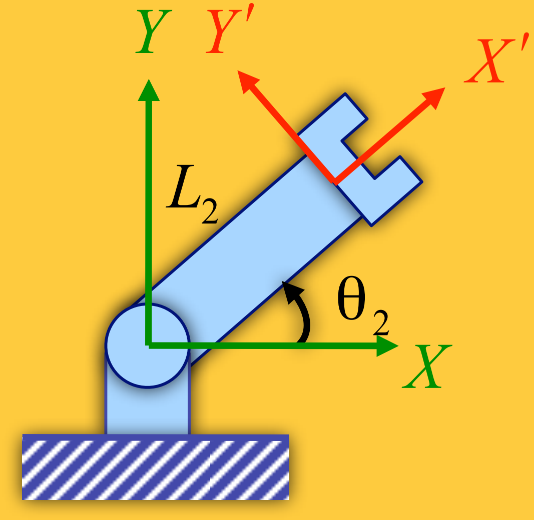


$$T = (rot\theta_1)(transl_1)(rot\theta_2)\boxed{(transl_2)}$$

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

Thinking of Transformations

In a view of global coordinate system

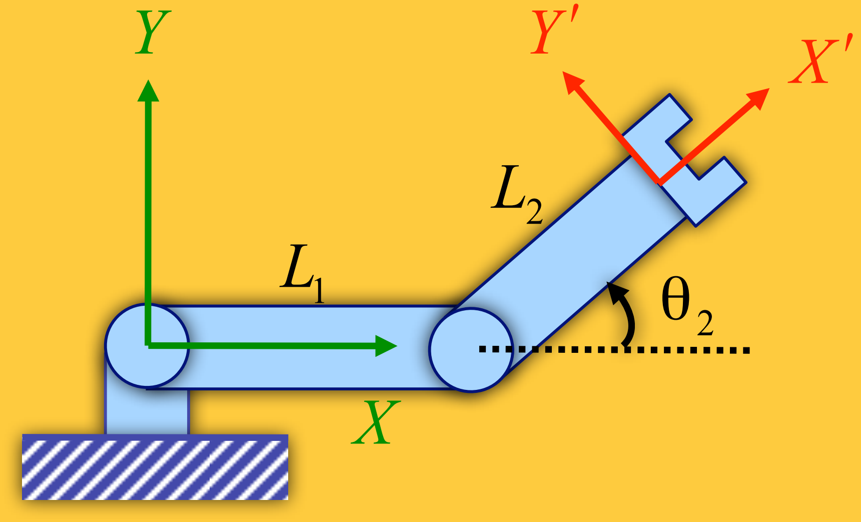


$$T = (rot\theta_1)(transl_1)\boxed{(rot\theta_2)(transl_2)}$$

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

Thinking of Transformations

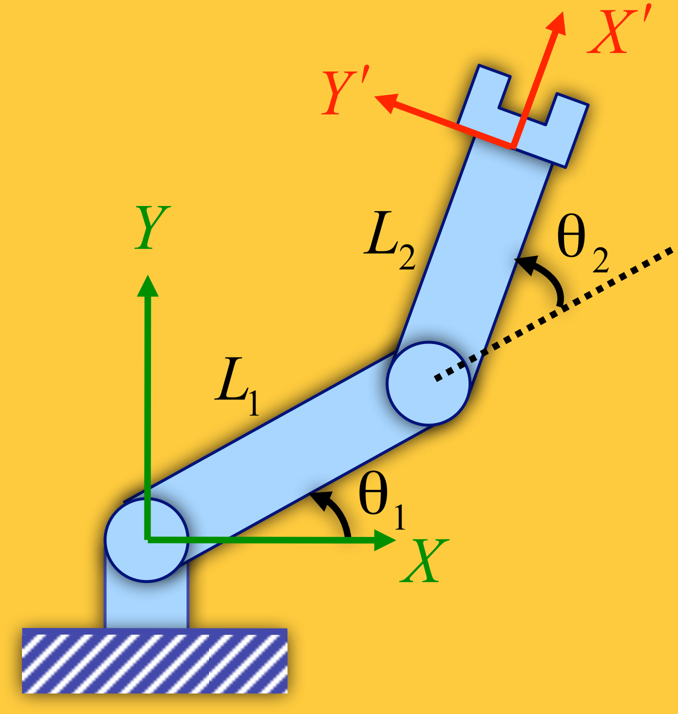
- In a view of global coordinate system



$$T = (rot\theta_1)\boxed{(transl_1)(rot\theta_2)(transl_2)}$$

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

Thinking of Transformations

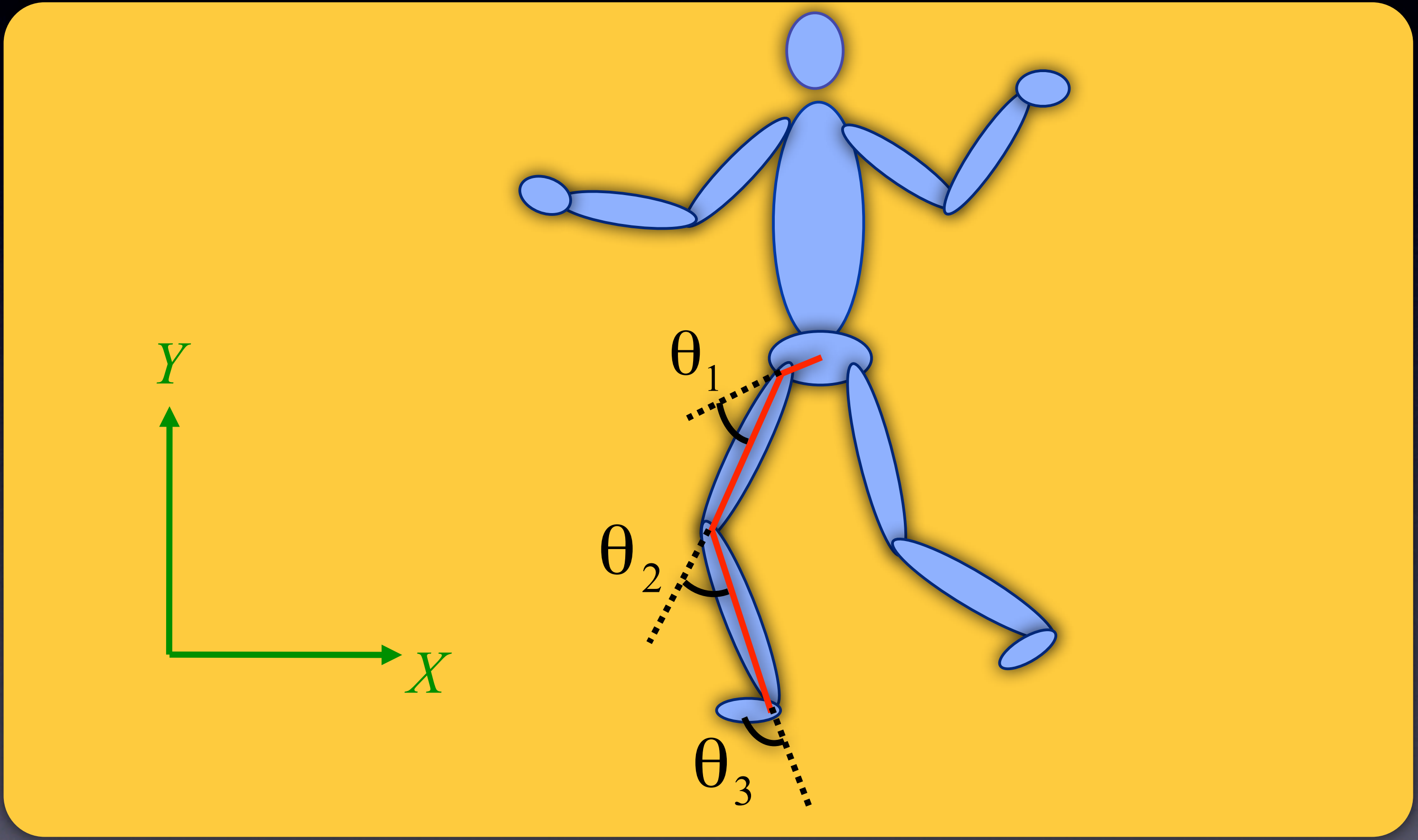


$$T = \boxed{(rot\theta_1)(transl_1)(rot\theta_2)(transl_2)}$$

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

Floating Base

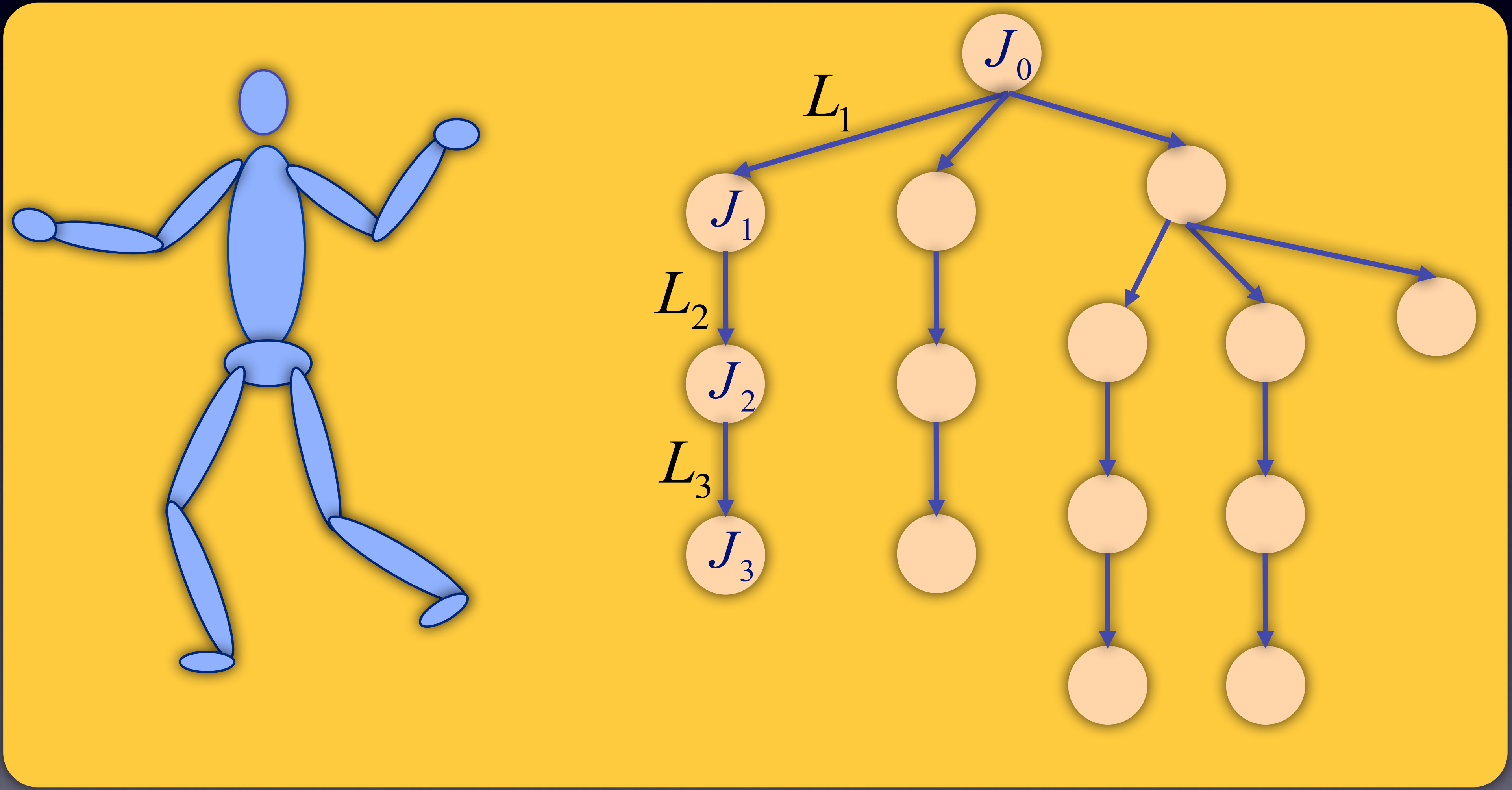
The position and orientation of the root segment are added



Representing Hierarchical Models

A tree structure

- A node contains a joint transformation
- A arc contains a link transformation



DOFs

Degree of Freedom (DOF): A variable φ describing a particular axis or dimension of movement within a joint

- Joints typically have around 1-6 DOFs ($\varphi_1 \ldots \varphi_N$) Can have more (up to 9 for affine)

- Changing the DOF values over time results in the animation of the skeleton

- Rigid body transformations: 6DOF
- Arbitrary rotations: 3DOF