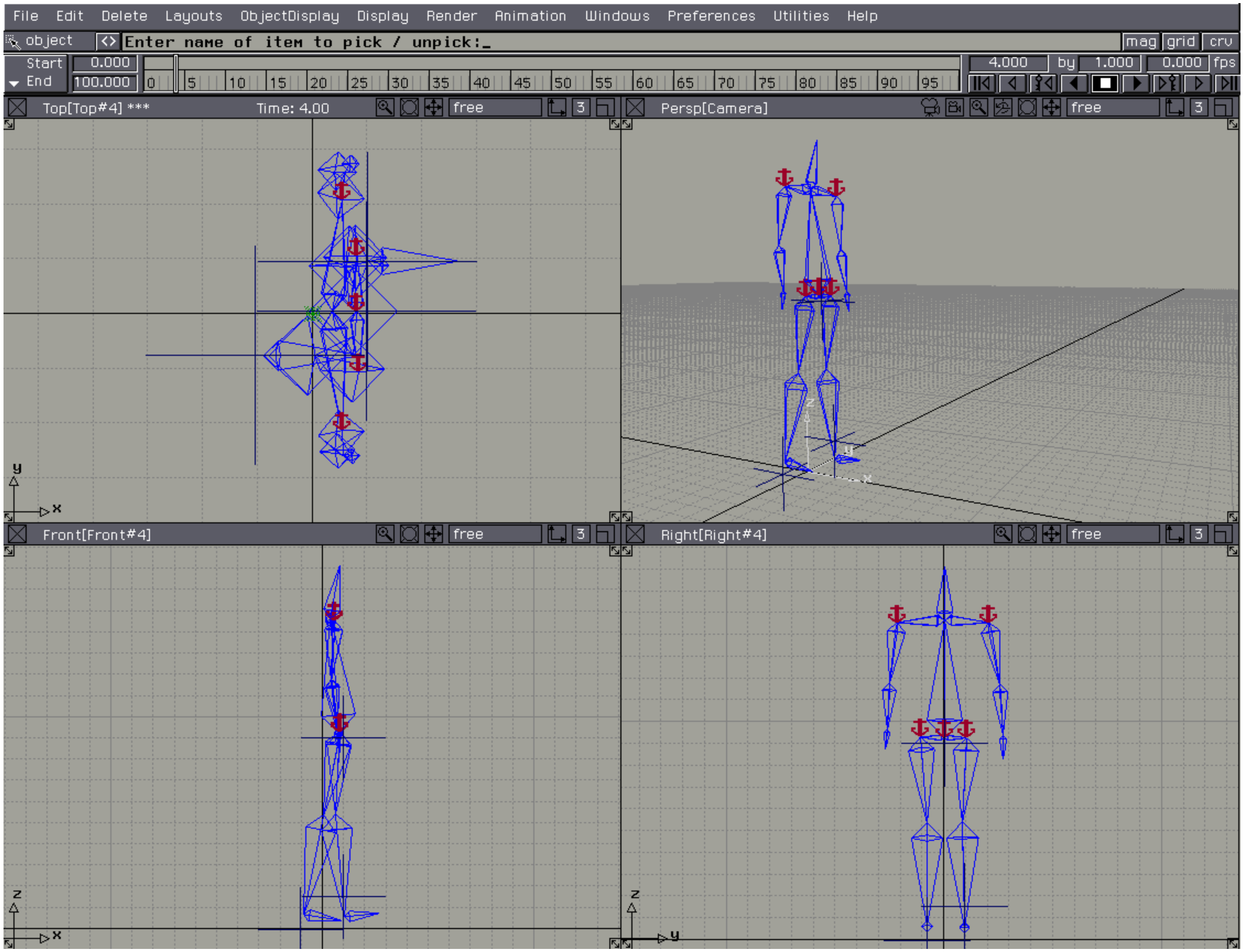
Keyframing

- fine level of control
- quality of motion depends on skill of animator



Keyframing

iterate:

- adjust trajectory
- play back motion

parameter/representation

interpolation path

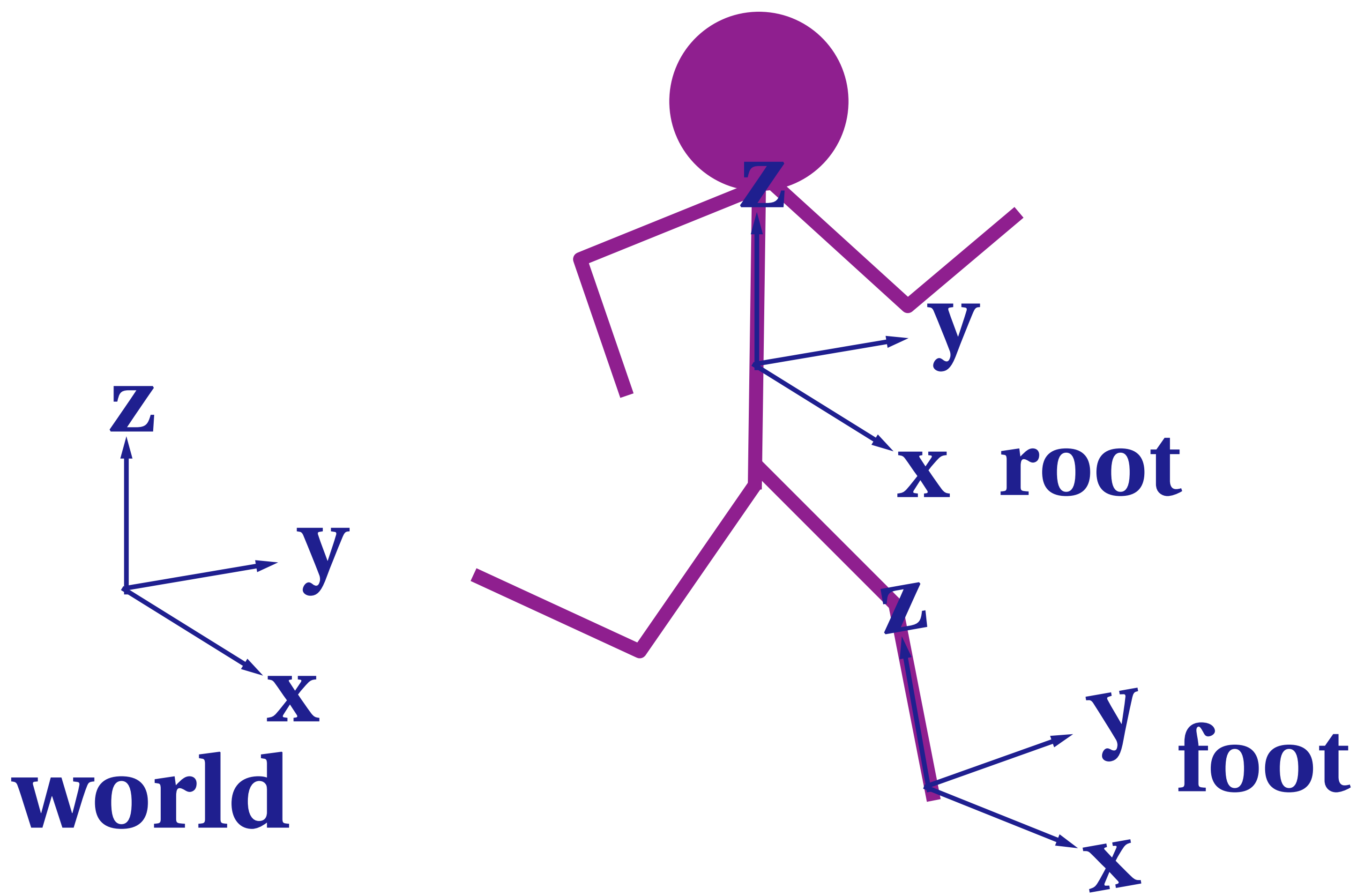
speed along path

Keyframing

parameters:

- locations/orientations
- joint angles
- shape -- flexible objects
- material properties (color, texture)
- camera motion (for animation)
- lighting

Coordinate Systems



Representation--orientation

transformation matrices

$$\begin{bmatrix} x_x & y_x & z_x & p_x \\ x_y & y_y & z_y & p_y \\ x_z & y_z & z_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Euler angles

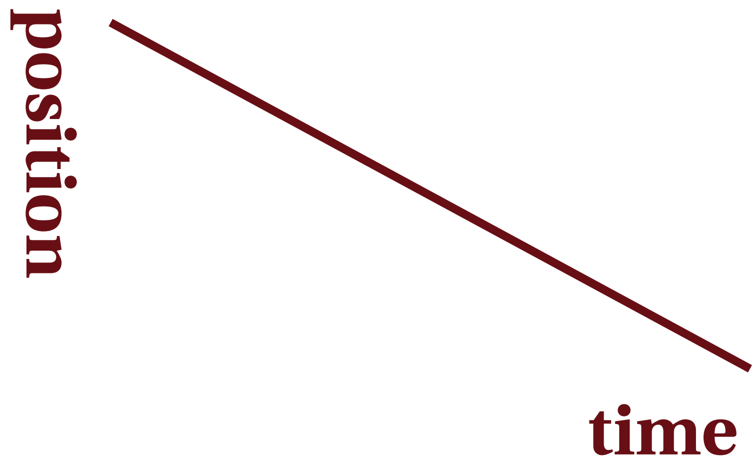
rot θ about x,
rot α about y...

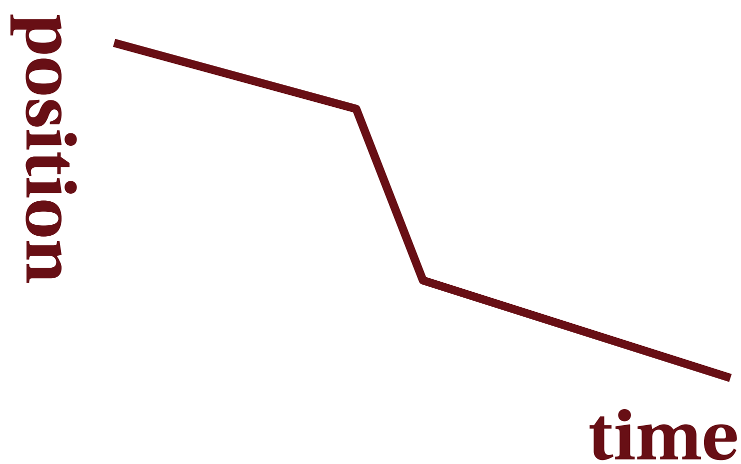
quaternions

(x,y,z) sin (θ/2), cos (θ/2)

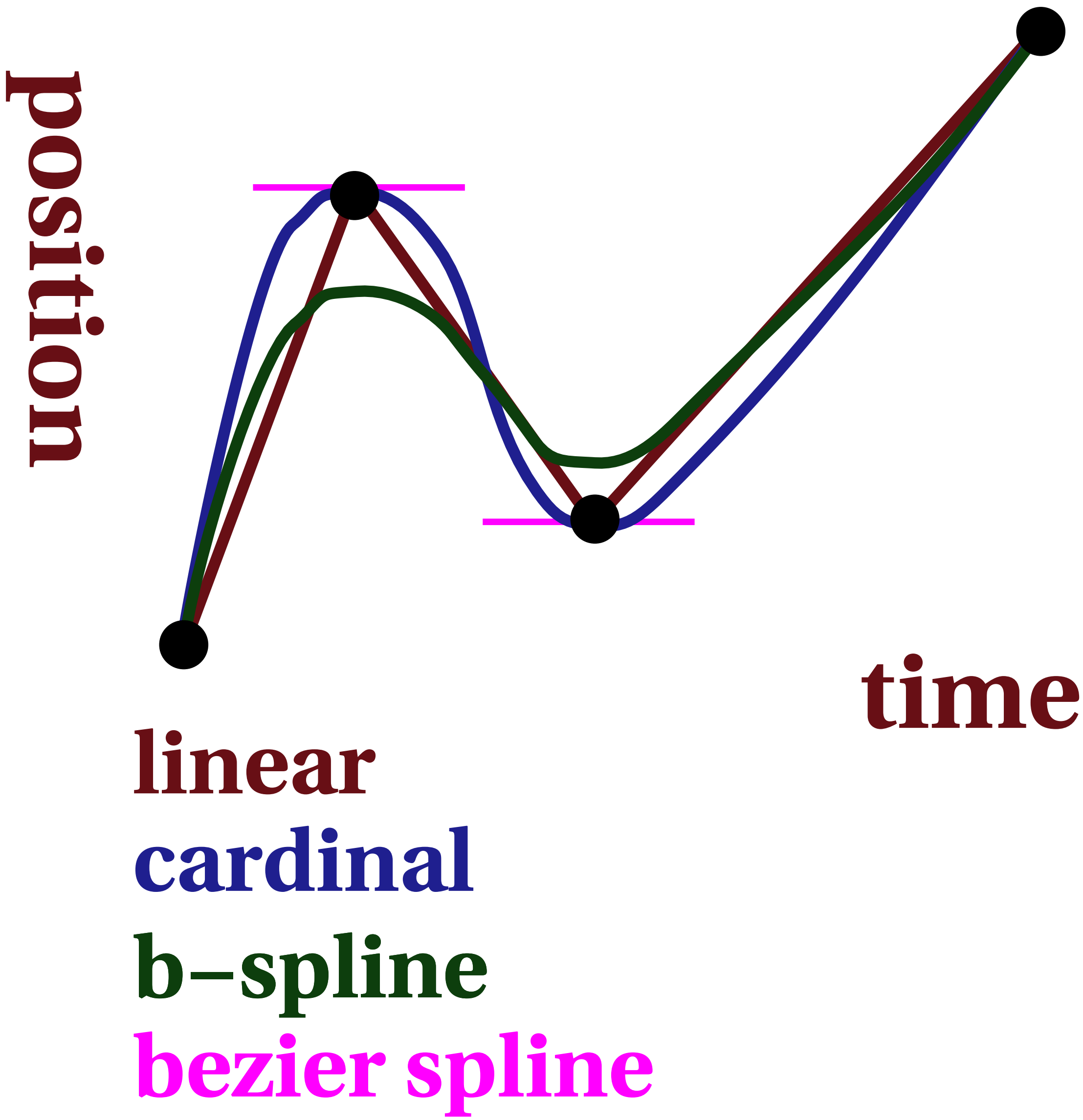
Keyframing -- Interpolation

Linear

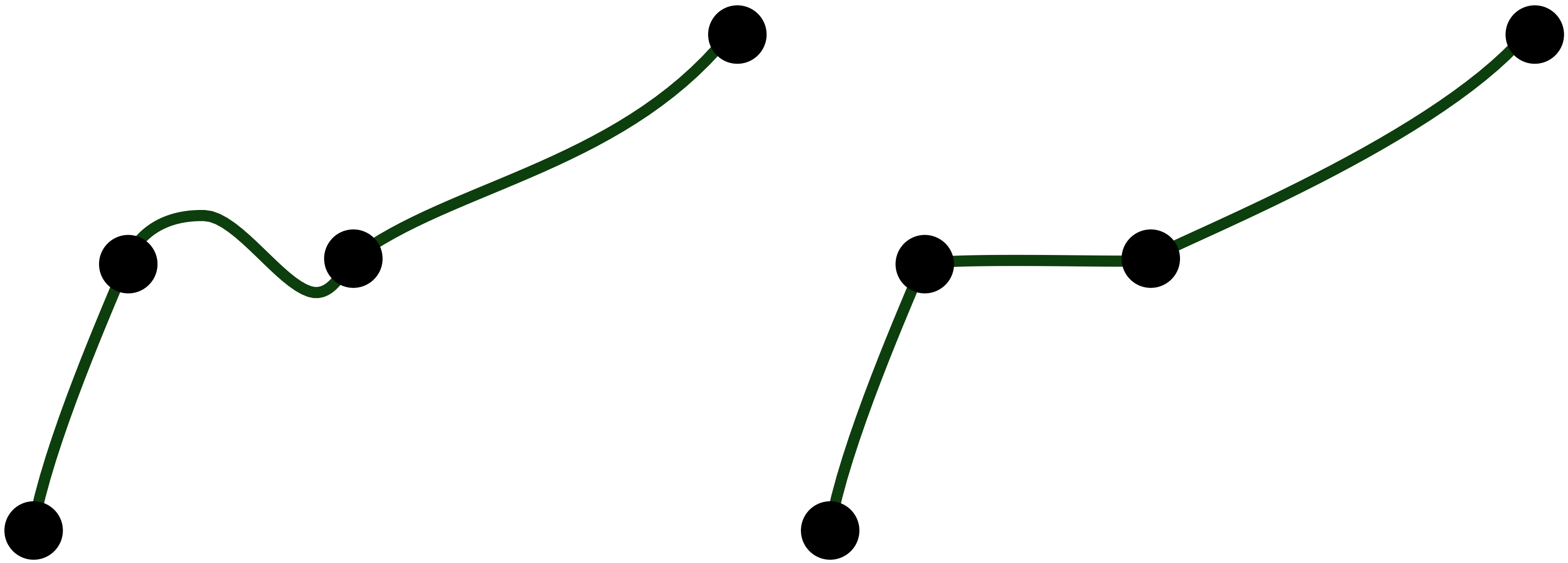




Splines



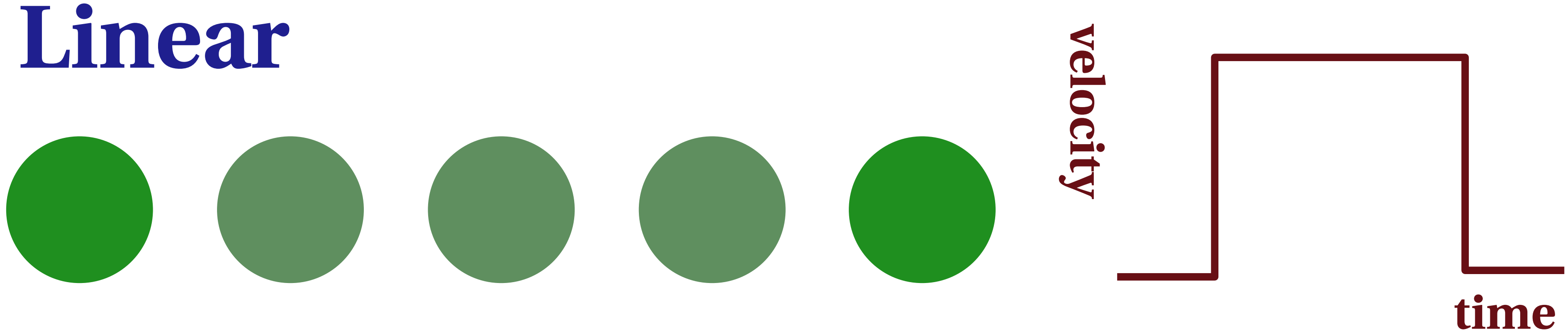
Splines



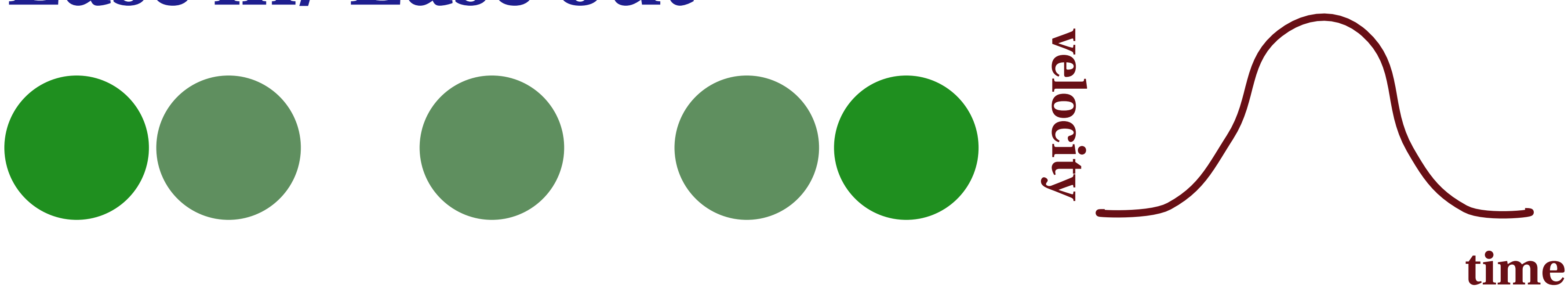
unintended wiggles

Keyframing
Velocity along path

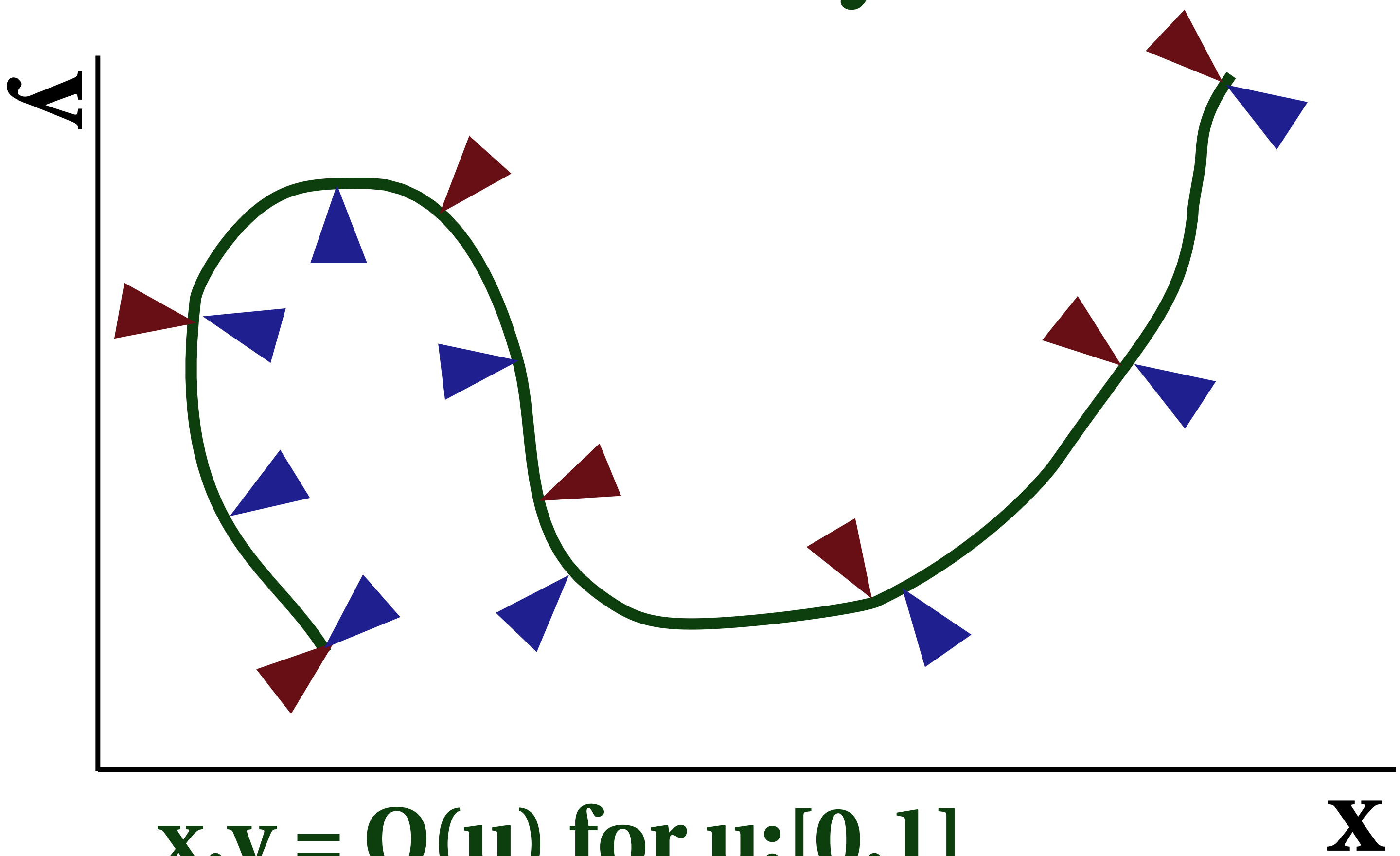
Linear



Ease in/ Ease out



Control of Velocity



x,y = Q(u) for u:[0,1]

equal arc lengths

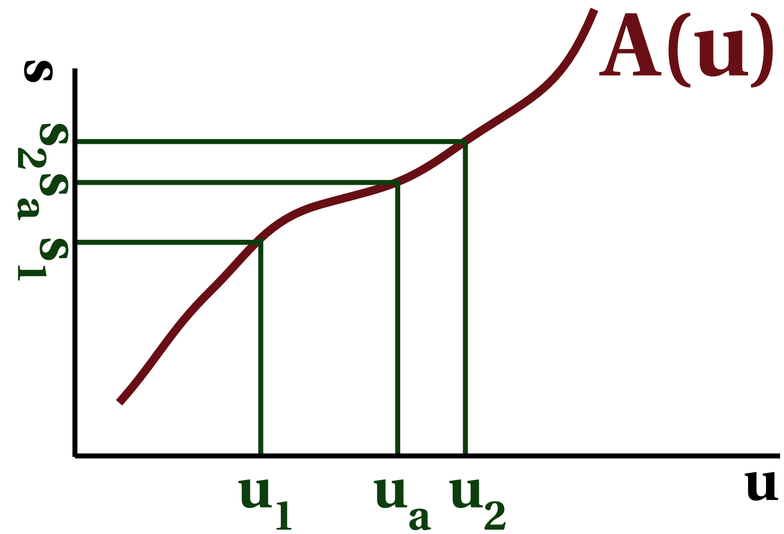
equal spacing in u

Arc–length reparametrization

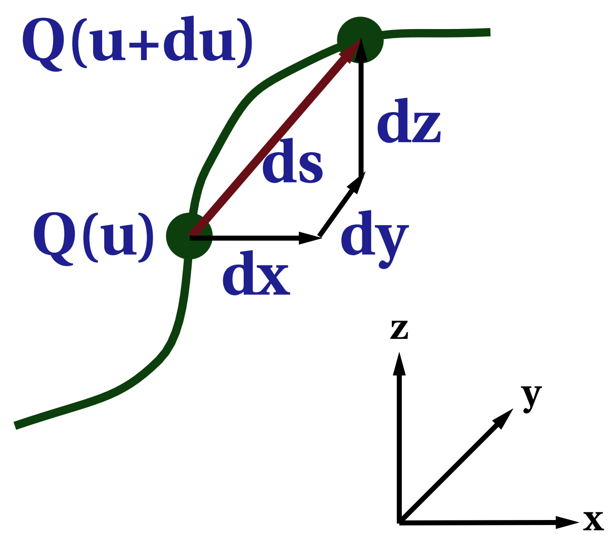
$s = A(u)$ where s is arc length

reparam: $Q(u)$ to $Q(A^{-1}(s))$

need to find $u = A^{-1}(s)$

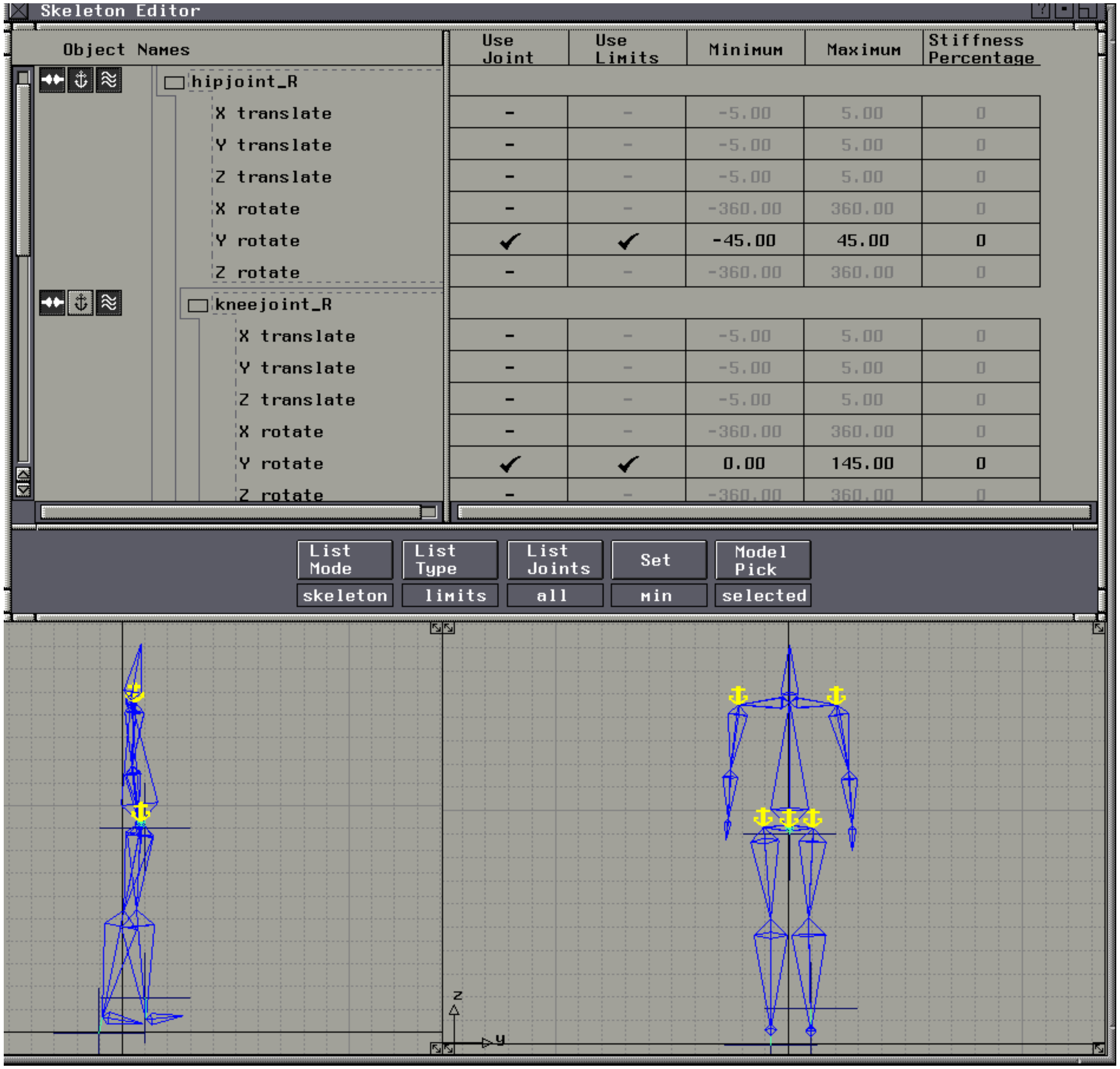


bisection search for a value of u where $A(u) = s$ with a numerical evaluation of $A(u)$ (details in Watt and Watt)



Keyframing -- Constraints

- Inverse Kinematics
- Joint Limits
- Position Limits



Control for the Animator

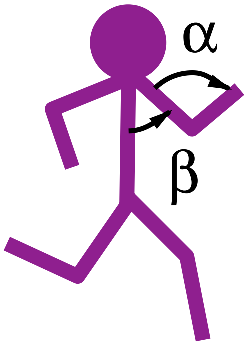
picking keyframe positions

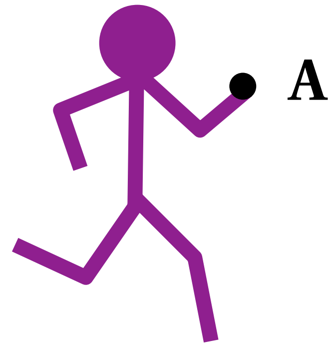
editing motion curves

control over velocity (timing)

specification of constraints

Kinematics –– the study of motion without regard to the forces that cause it.





Forward: $A = f(\alpha,\beta)$

draw graphics

Inverse: $\alpha,\beta = f^{-1}(A)$

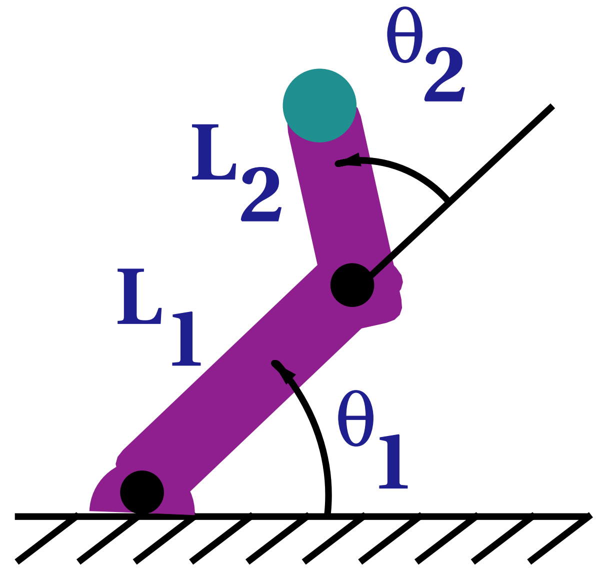
specify fewer degrees of freedom

more intuitive control of dof
- pull on hand
- glue feet to the ground

Forward Kinematics

$$x = L_1 \cos\theta_1 + L_2 \cos(\theta_1 + \theta_2)$$

$$y = L_1 \sin\theta_1 + L_2 \sin(\theta_1 + \theta_2)$$



$$\begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} \quad \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

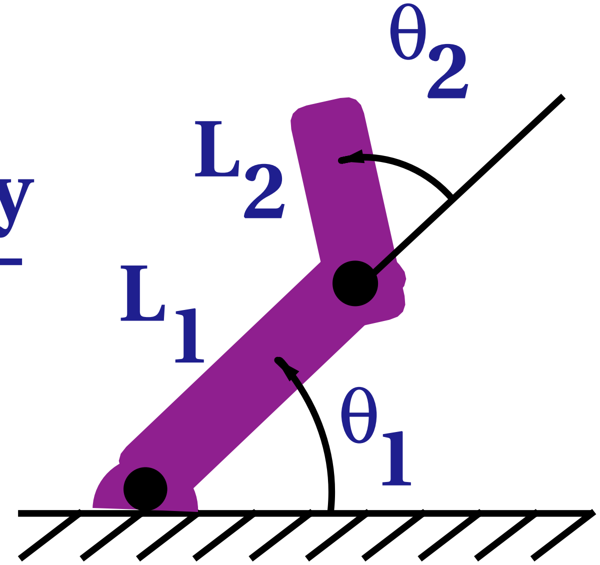
$$\begin{bmatrix} \quad \end{bmatrix} = \begin{bmatrix} \text{rot } \theta_1 \end{bmatrix} \begin{bmatrix} \text{trans } L_1 \end{bmatrix} \begin{bmatrix} \text{rot } \theta_2 \end{bmatrix} \begin{bmatrix} \text{trans } L_2 \end{bmatrix}$$

Inverse Kinematics

$$\theta_2 = \frac{\cos\ (x^2 + y^2 - L_1^2 - L_2^2)}{2\,L_1\,L_2}$$

$$\theta_1 = \frac{-(L_2 \sin\theta_2)x + (L_1 + L_2 \cos\theta_2)y}{(L_2 \sin\theta_2)y + (L_1 + L_2 \cos\theta_2)x}$$

$$\theta = f^{-1}(x)$$



What makes IK hard?

- redundancy
- singularities
- goal of "natural looking" motion

Danse Interactif and Improv – Perlin, NYU

Keyframed actions

Additive noise