Animation

- moving things around
- 2D, 3D or a mixture
- 15–30 fps, consistent
- double buffering

Behaviors (3 lectures):

- low level (walking, hitting)
- collision detection
- transitions between behaviors
- higher level (camera, path planning)
- group behaviors

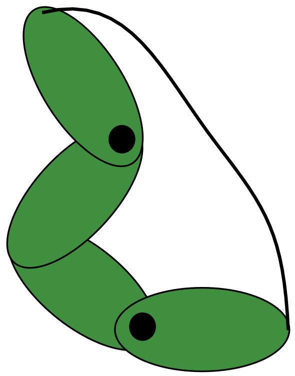
Generating Motion

What Matters?

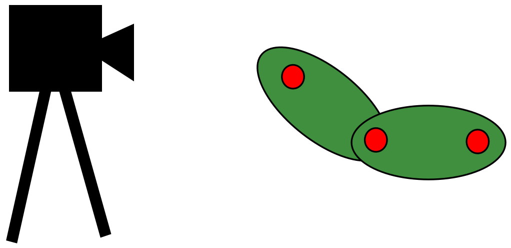
- personality
- skills
- controllable/AI
- quality of motion appropriate for rendering style

Generating Motion

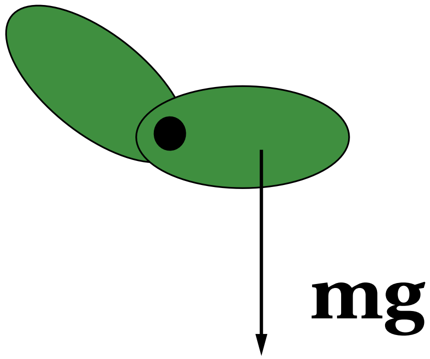
keyframing



recording motion data

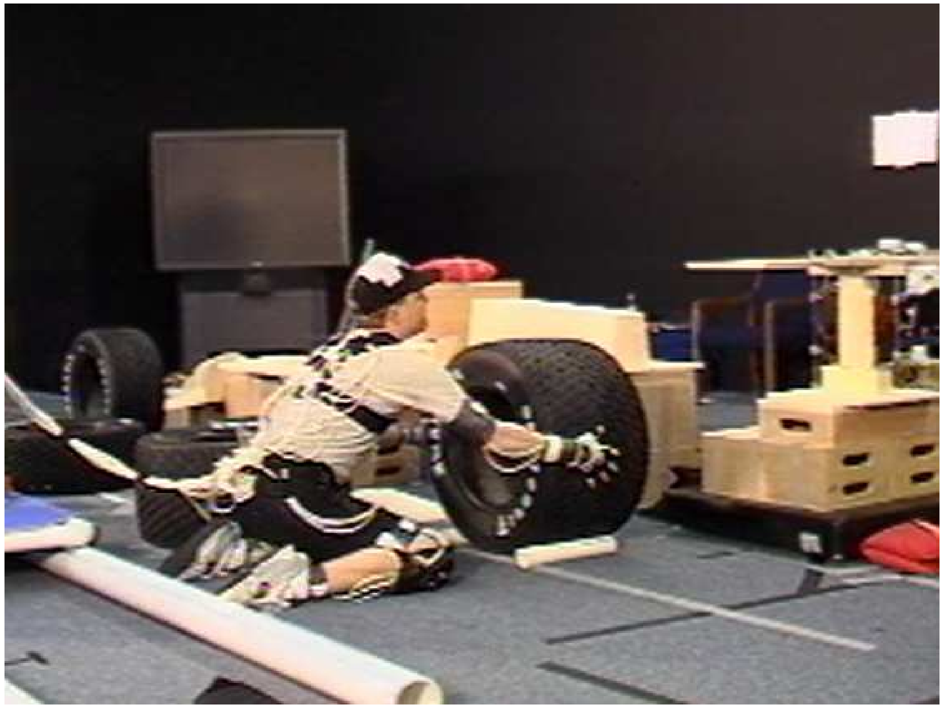


simulation
(procedural)



Motion Capture

natural–looking motion
hard to generalize motions
registration is difficult



images courtesy of the Microsoft Motion Capture Group

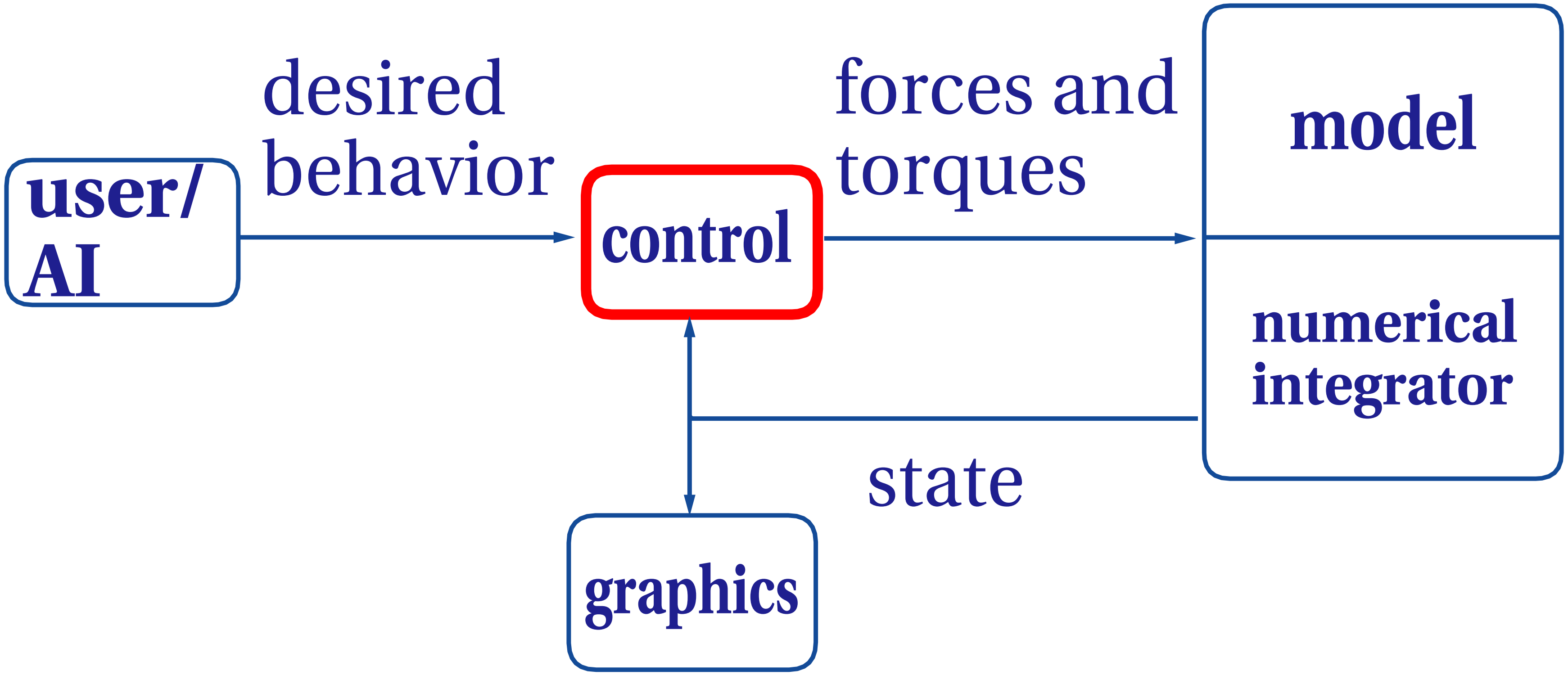
Simulation (broadly defined)

physics is hard

pseudo–physics is somewhat hard

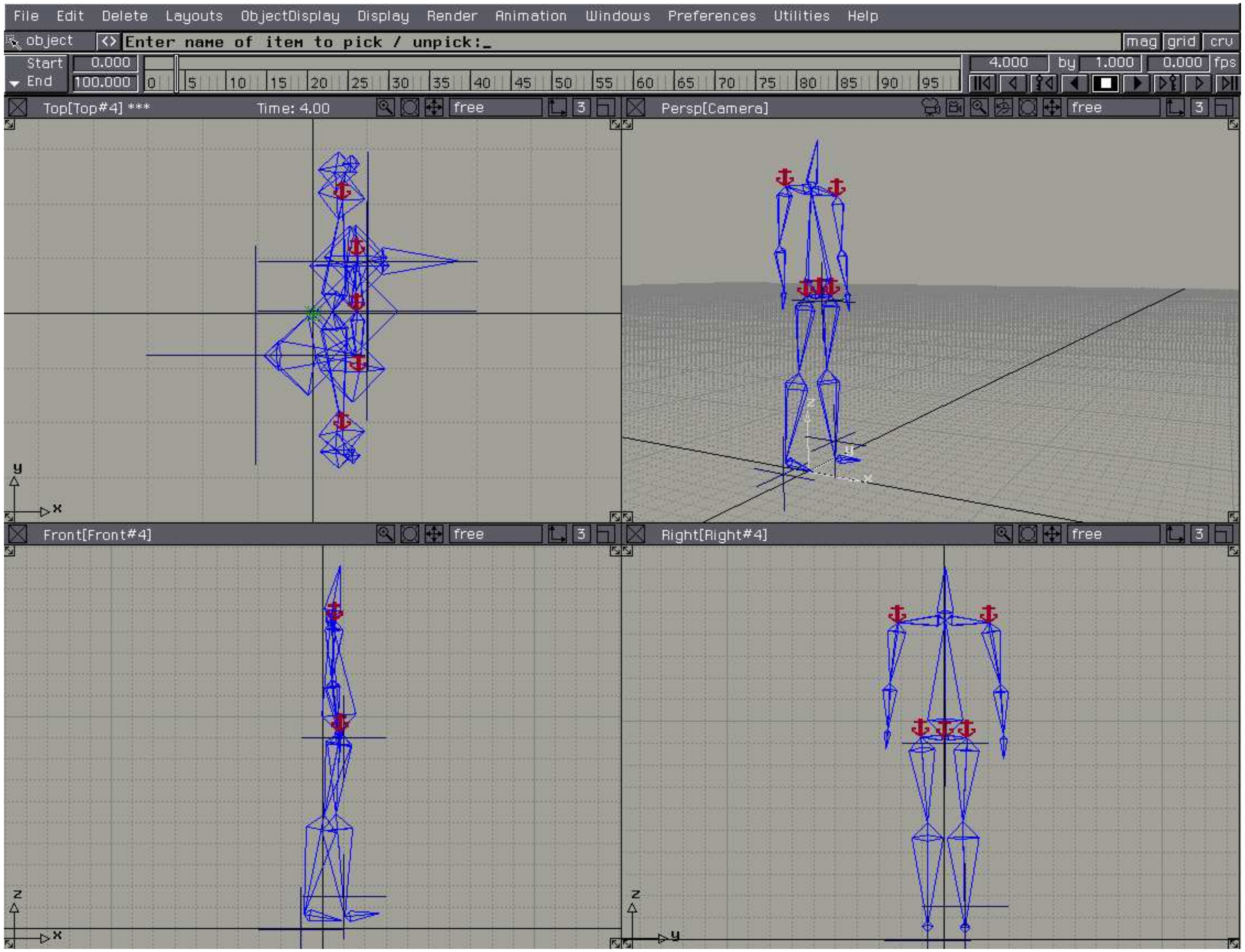
control is very hard

generalization



Keyframing

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



Keyframing

iterate:

adjust trajectory

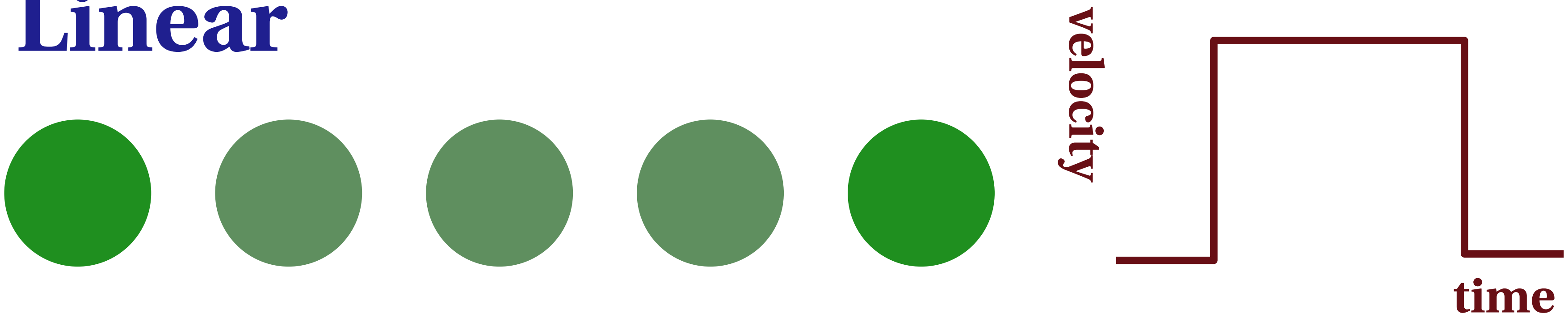
play back motion

parameters:

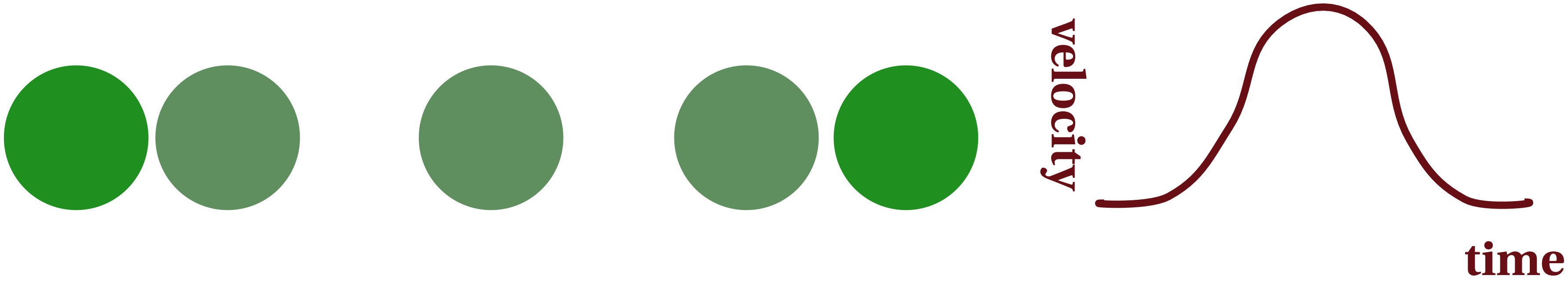
- locations
- joint angles
- shape -- flexible objects
- material properties (color, texture)
- camera motion (for animation)
- lighting

Keyframing –– Interpolation Inbetweening

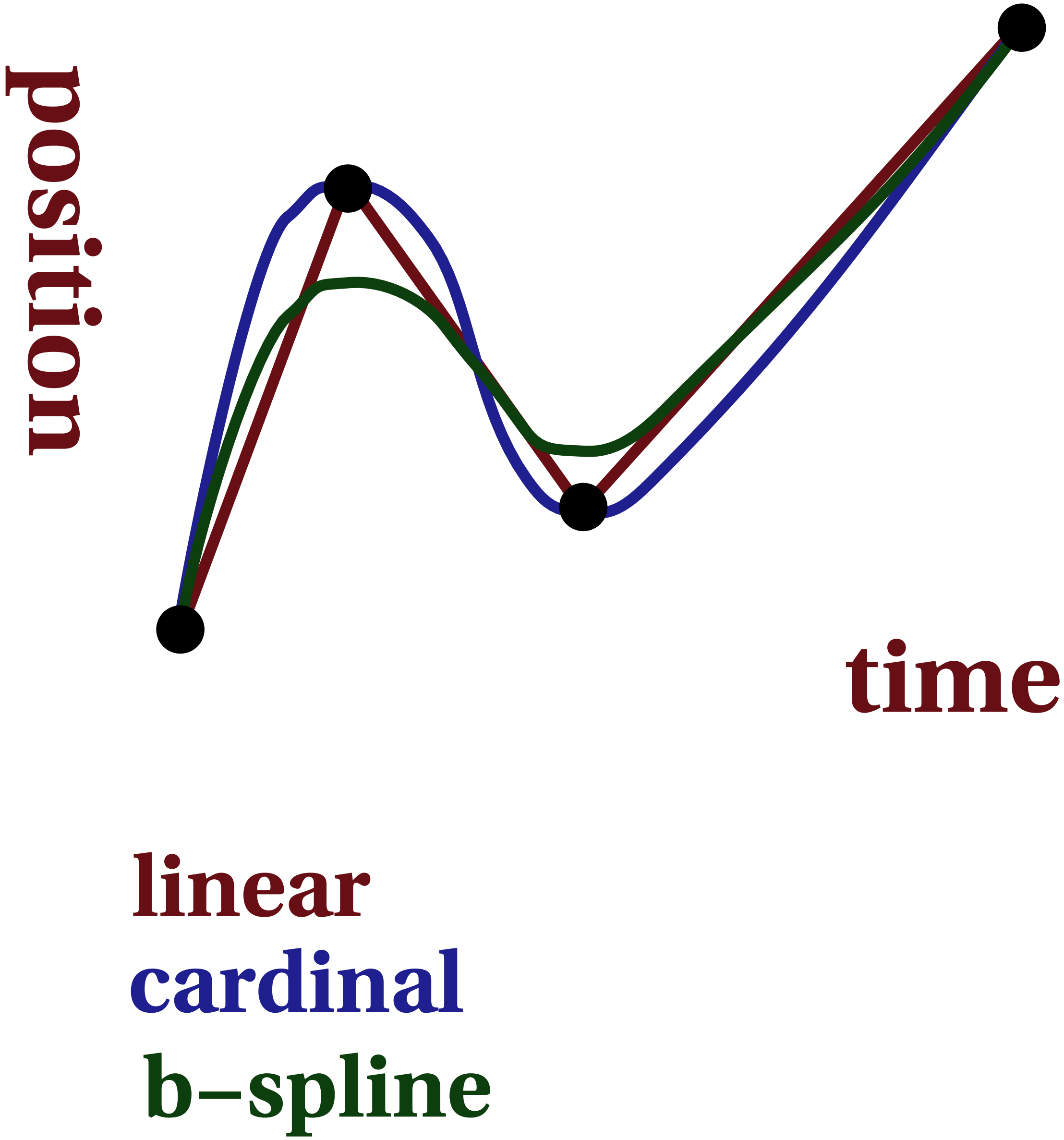
Linear



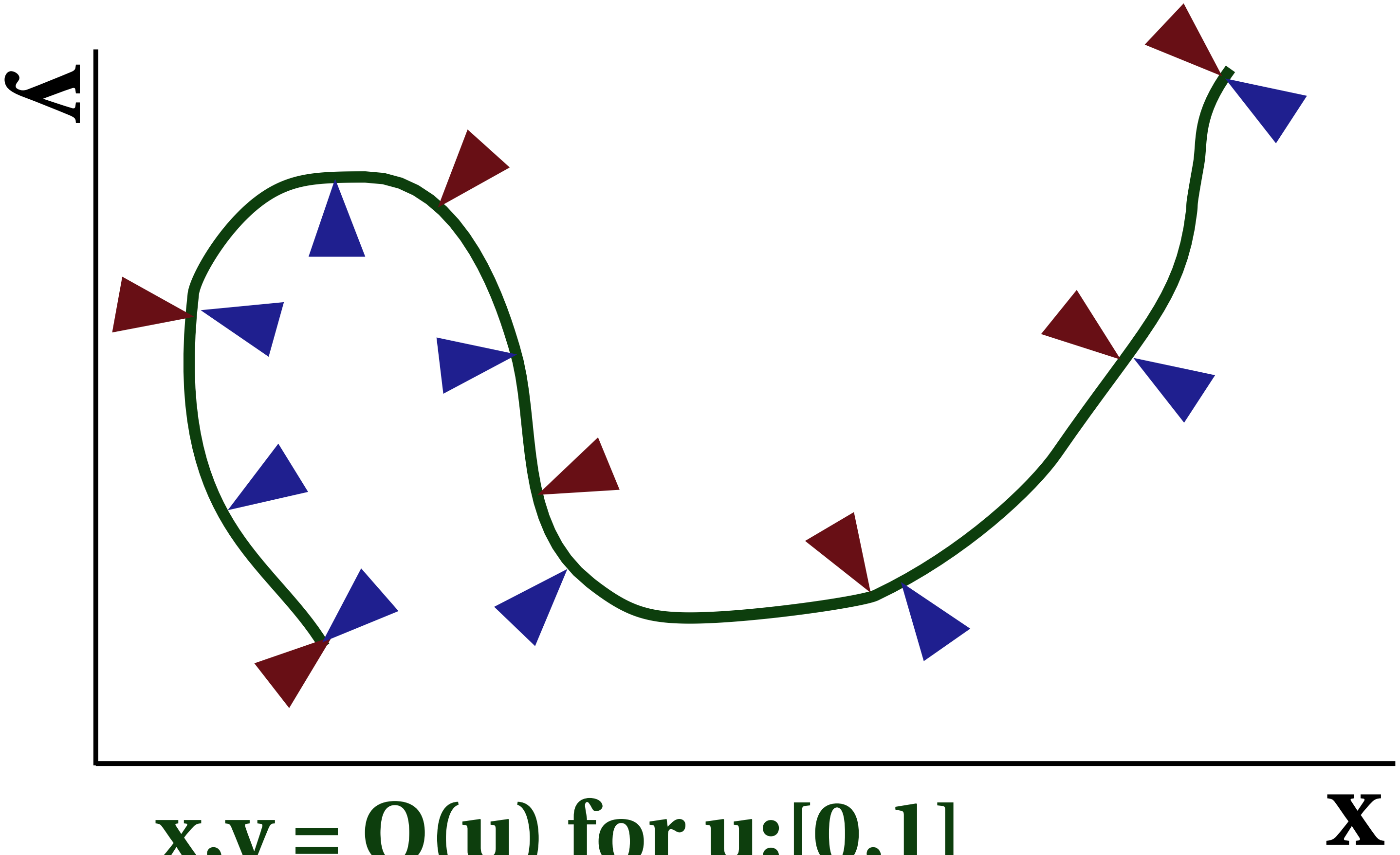
Ease in/ Ease out



Spline-driven Animation



Spline-driven Animation



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

equal arc lengths

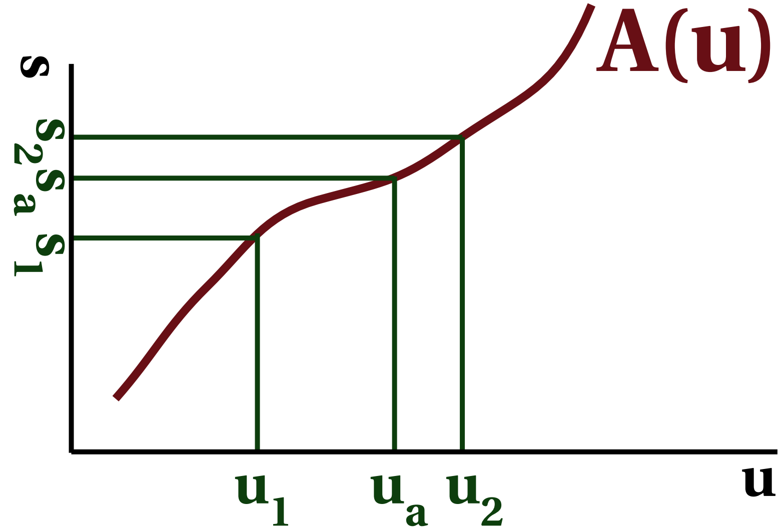
equal spacing in u

Arc–length reparametrization

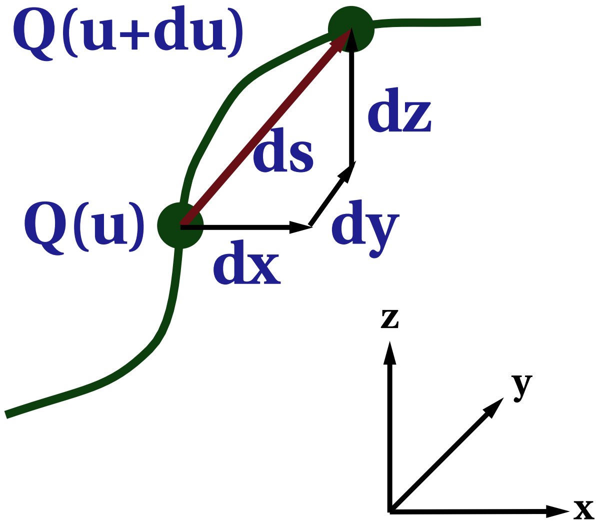
$s = A(u)$ where s is arc length

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

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

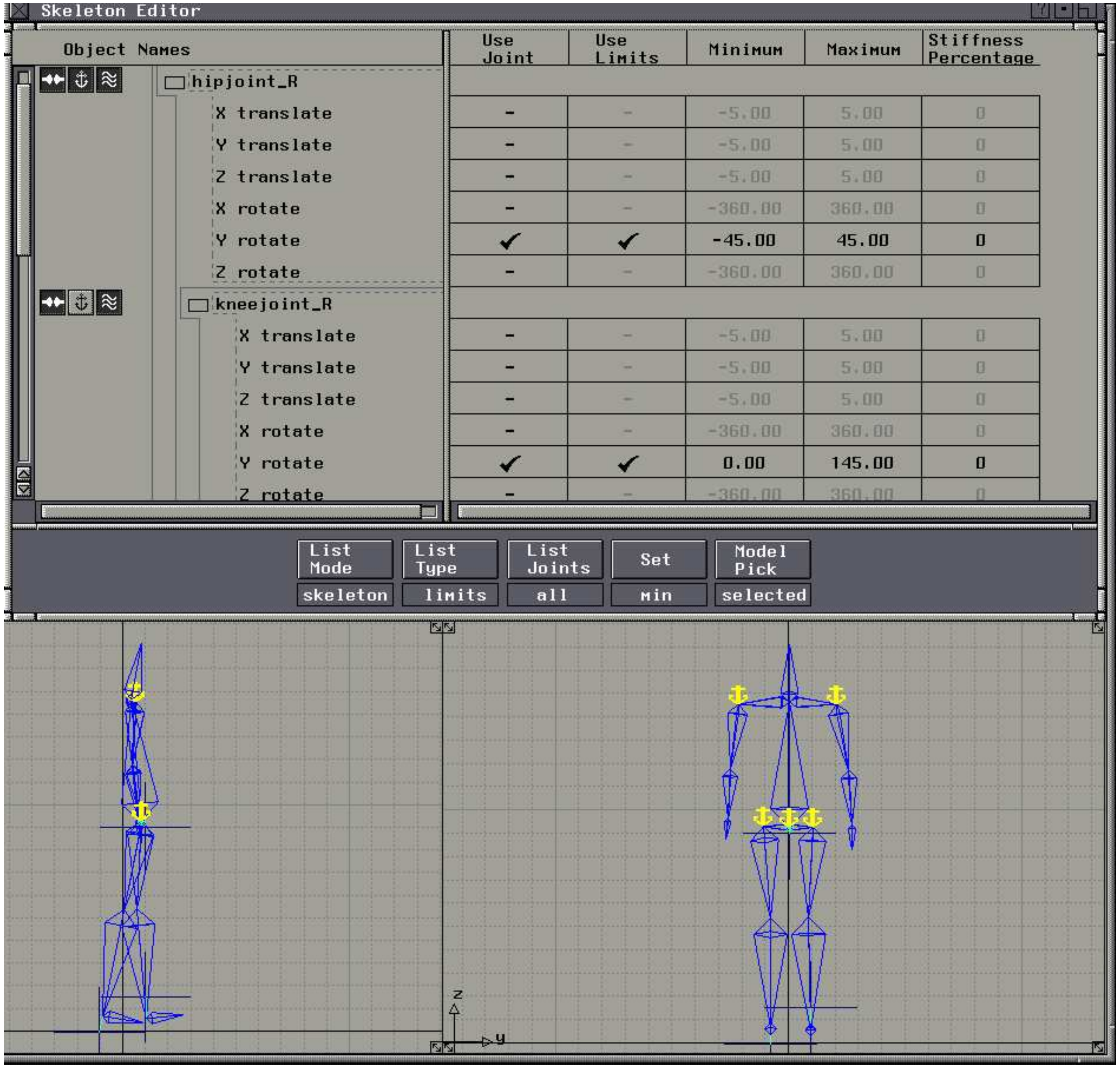


bisection search for a value of u where $A(u) = s$ with a numerical evaluation of $A(u)$ (details in Watt and Watt)

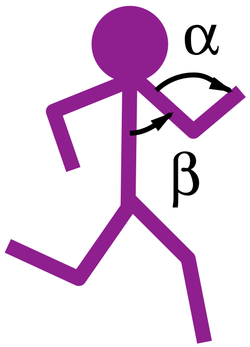


Keyframing -- Constraints

- Inverse Kinematics
- Joint Limits
- Position Limits

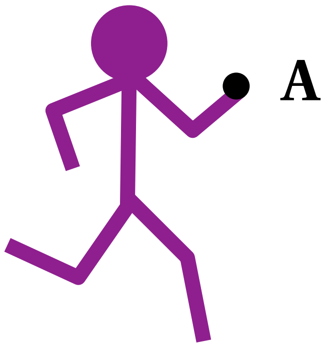


Kinematics –– the study of motion without regard to the forces that cause it.



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

draw graphics



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

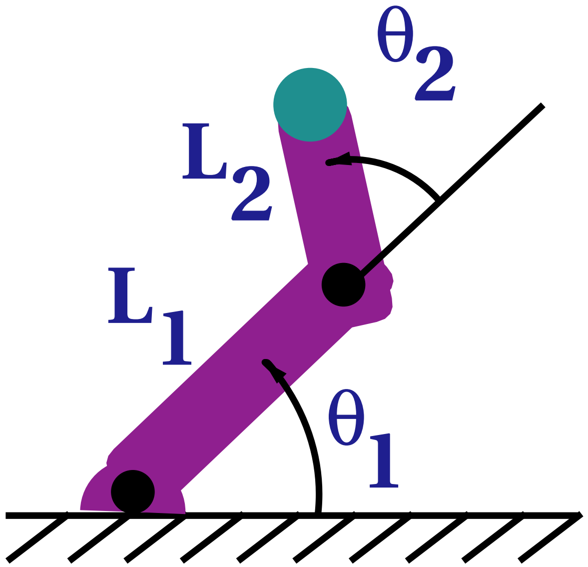
specify fewer degrees of freedom

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

Forward Kinematics

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

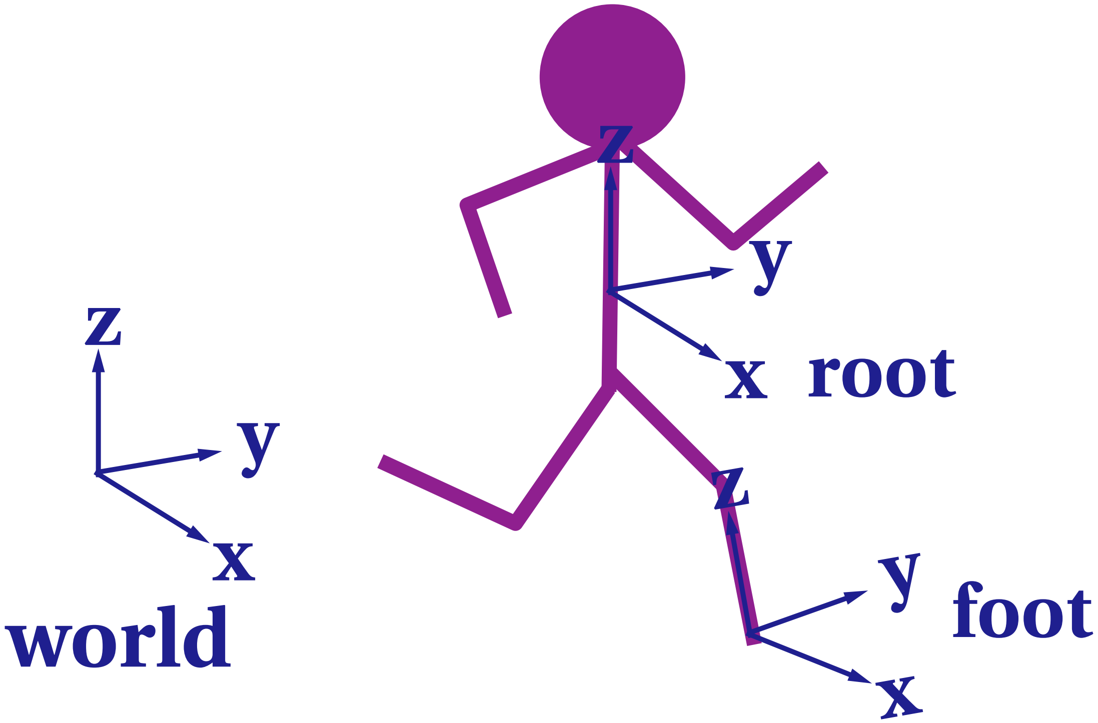
$$y = L_1 \sin\theta_1 + L_2 \sin(\theta_1 + \theta_2)$$



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

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

Coordinate Systems



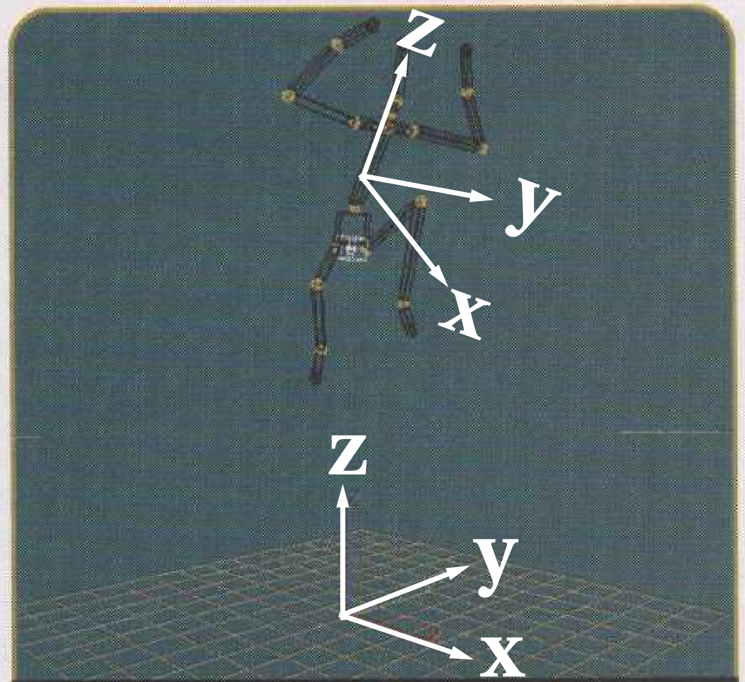
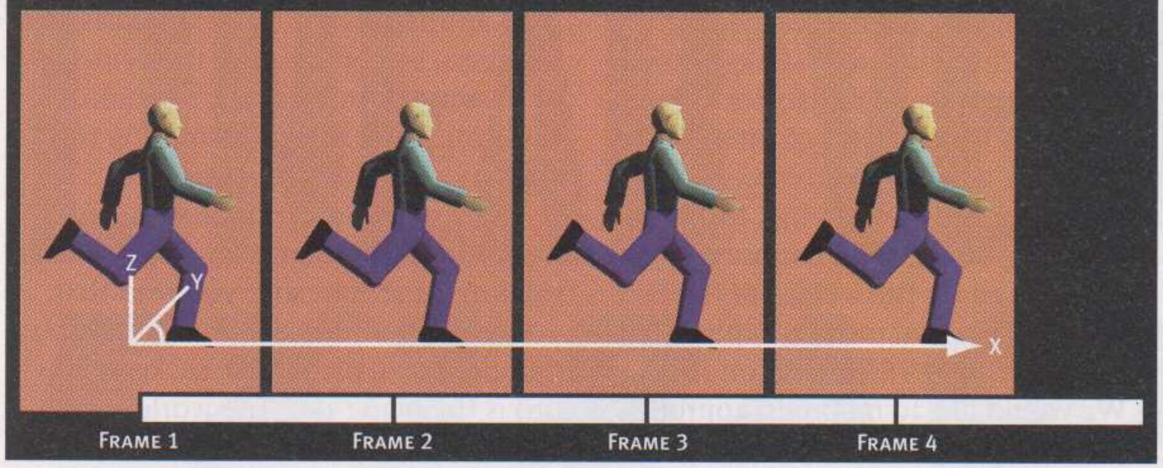


FIGURE 3. *Zed in an animated jump. Z-translation is controlled by the animation. The Base is still on the floor.*

FIGURE 1. *As animation progresses, the root moves away from the origin.*

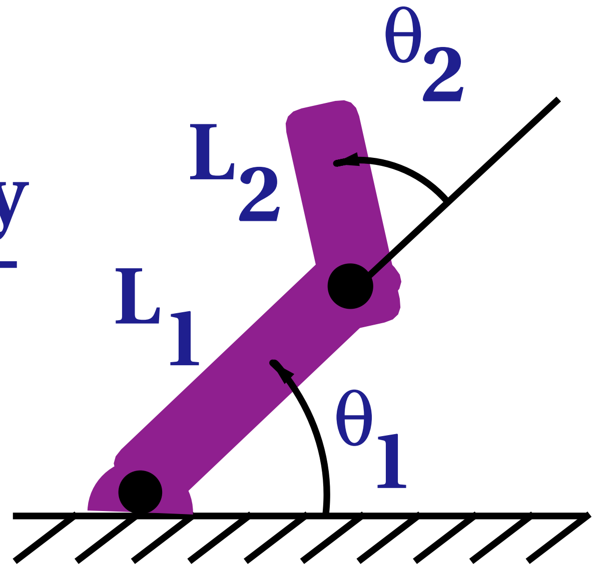


Inverse Kinematics

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

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

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



What makes IK hard? -- many dof

$$\begin{bmatrix} x_x & y_x & z_x & p_x \\ x_y & y_y & z_y & p_y \\ x_z & y_z & z_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} a_x & b_x & c_x & d_x \\ a_y & b_y & c_y & d_y \\ a_z & b_z & c_z & d_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

12 equations, 6 unknowns ($\theta_1,\ldots\theta_6$)

only 3 of the 9 rotation terms are independent

non–linear, transcendantal equations

What makes IK hard? -- Redundancy

Add constraints to reduce redundancies

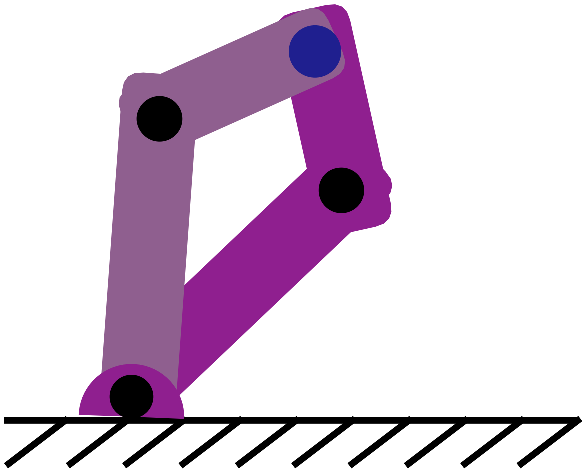
Choose solution that is

"closest" to current configuration

move outermost links the most

energy minimization

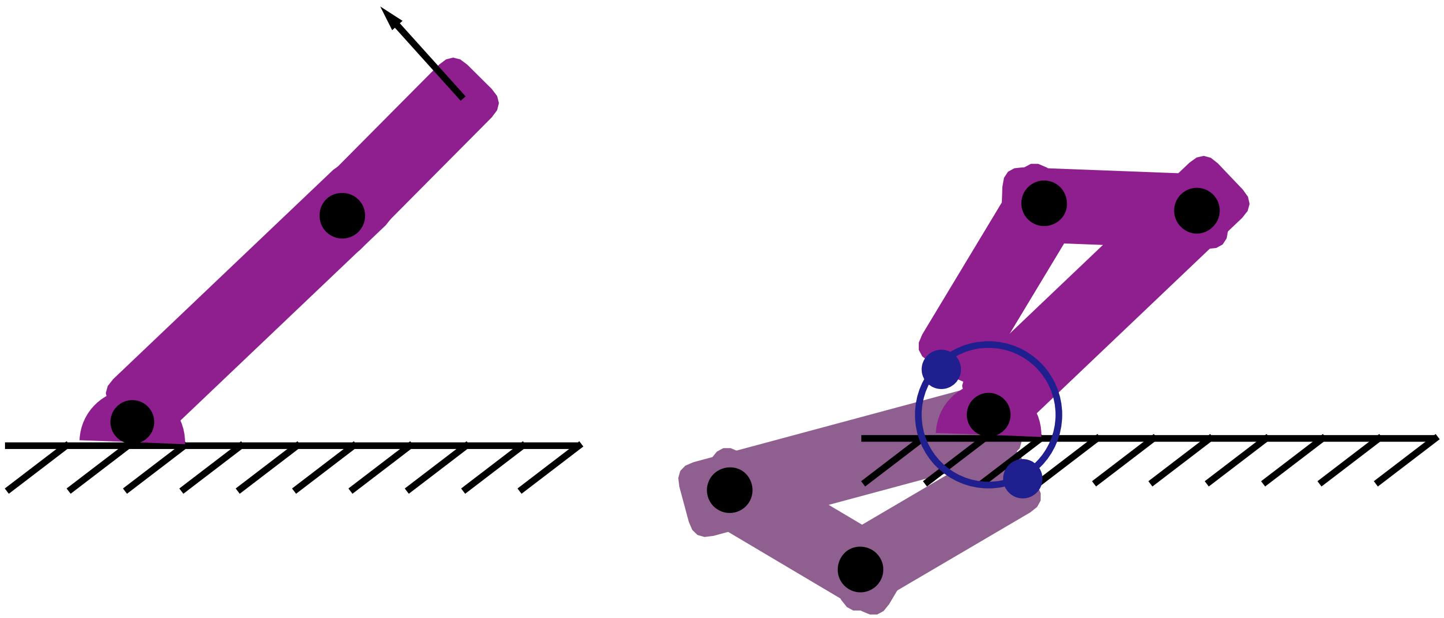
minimum time



What makes IK hard? -- singularities

ill–conditioned near singularities

high state space velocities for
low cartesian velocities



What makes IK hard?

- goal of "natural looking" motion
  - minimum jerk
  - equilibrium point trajectories

The Jacobian

$f(\theta) = x$ x is of dimension n (generally 6)

θ is of dimension m (# of dof)

Jacobian is the n x m matrix relating differential changes of θ ($d\theta$) to differential changes of x (dx)

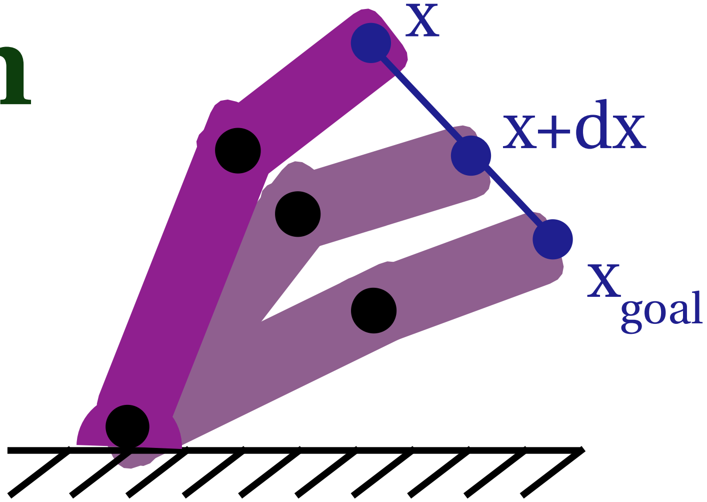
$J(\theta)\, d\theta = dx$ where the ijth element of J is

$$J_{ij} = \frac{\delta f_i}{\delta x_j}$$

Jacobian maps velocities in joint angle space to velocities in cartesian space

IK and the Jacobian

$$\theta = f^{-1}(x)$$
$$dx = J\, d\theta$$
$$d\theta = J^{-1}\, dx$$



Inverting the Jacobian

J is n x m–– not square in general
compute pseudo–inverse

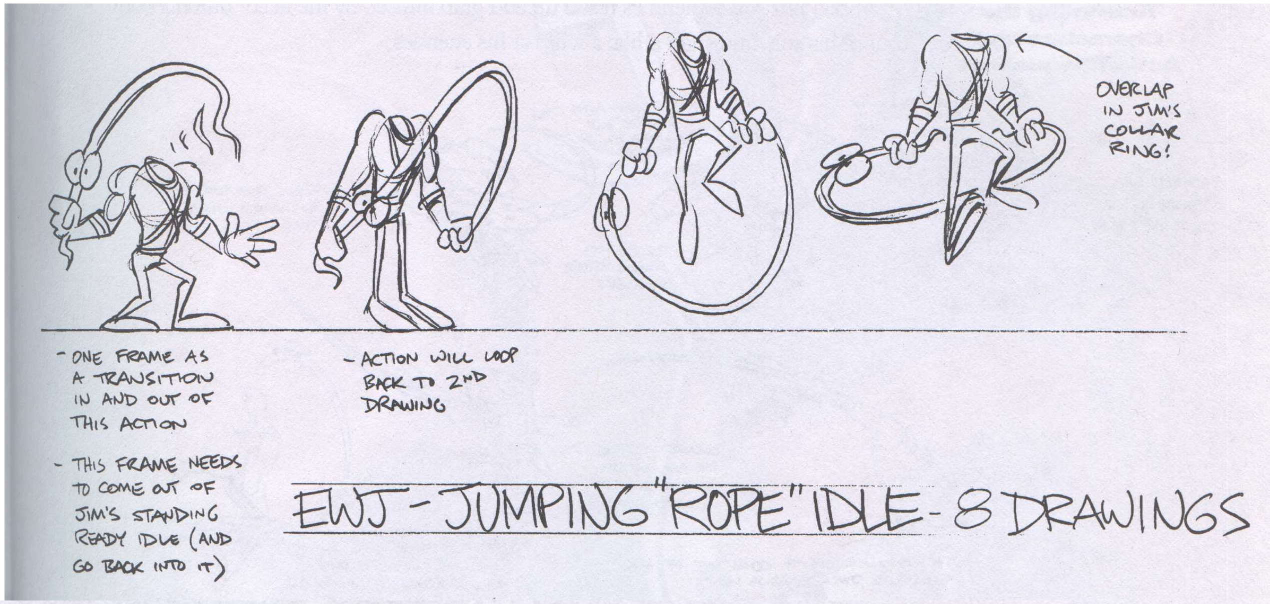
Singularities cause the rank of the Jacobian to change

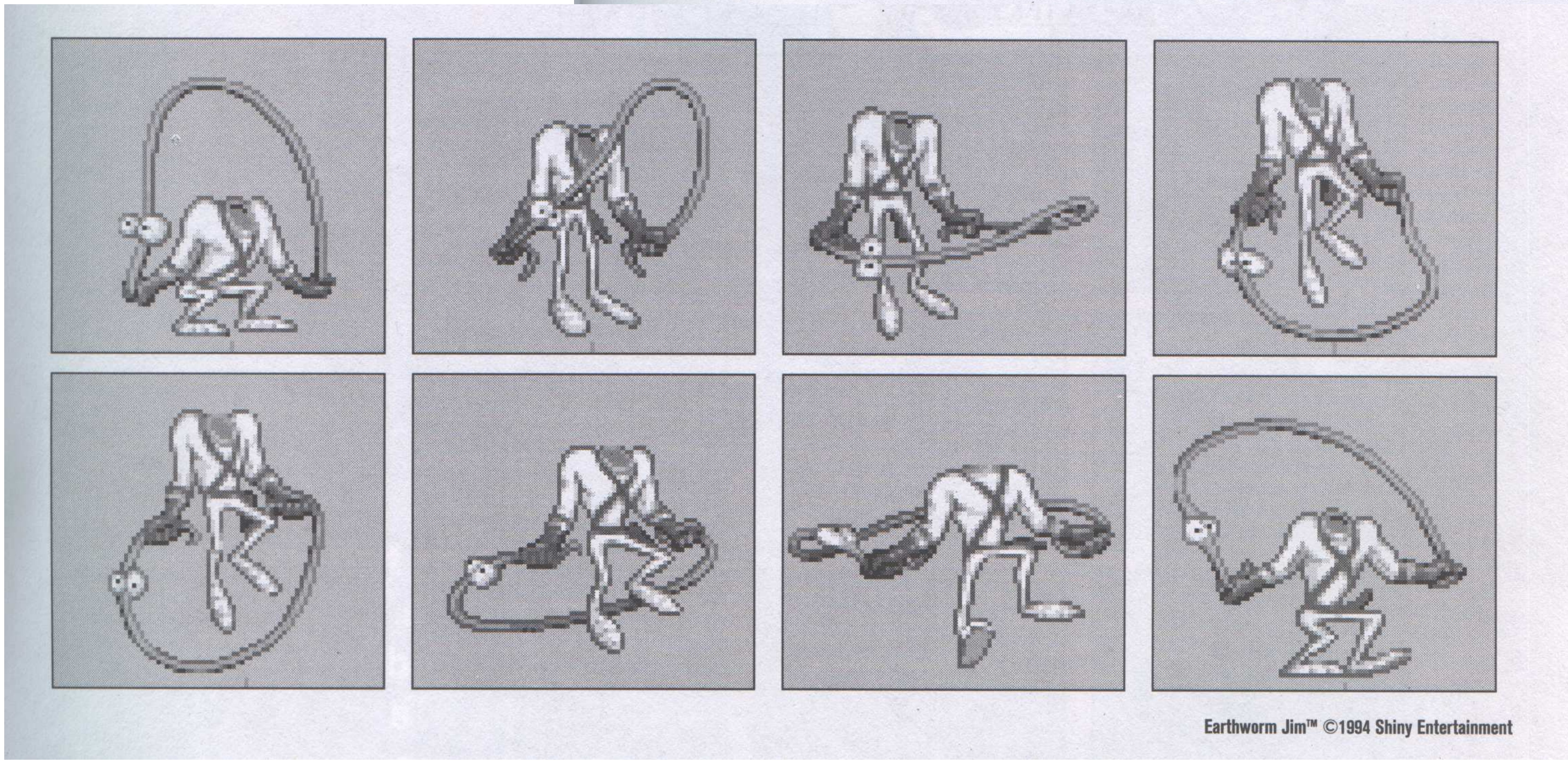
Damped Least Squares:
find solution that minimizes

$$||J - dx||^2 + \lambda^2 ||d\theta||^2$$

Hand Drawn Animation: 2D

- sketches
- pencil tests
- inking
- coloring
- digitize to sprites





Computer Animation: 2D or 3D

sketches

models and materials

key configurations

playback of motion
or render to sprites

Improv, Perlin, NYU