# Computer Animation

Michael Kazhdan

(601.457/657)

# Overview

- Some early animation history
  - http://web.inter.nl.net/users/anima/index.htm
  - http://www.public.iastate.edu/~rllew/chrnearl.html
- Computer animation

# Thaumatrope

Q: Why does animation work?

A: Persistence of vision

1824 John Ayerton invents the *thaumatrope*

Or, 1828 Paul Roget invents the *thaumatrope*

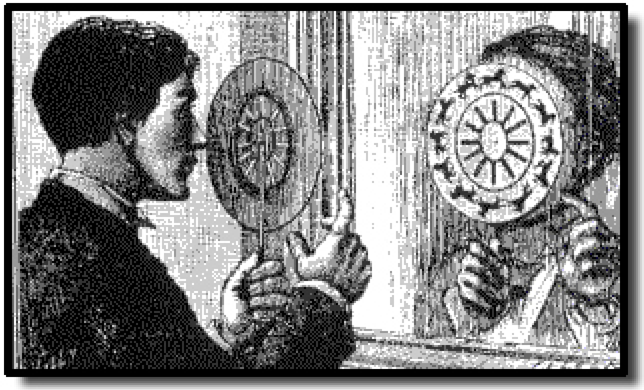

# Phenakistoscope

Invented independently by 2 people in 1832

Disc mounted on spindle

Viewed through slots with images facing mirror

Turning disc animates images



Aliasing as a feature, not a bug

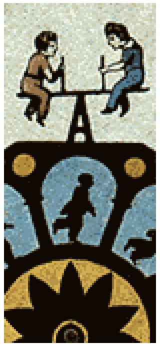
# Phenakistoscope

Invented independently by 2 people in 1832

Disc mounted on spindle

Viewed through slots with images facing mirror

Turning disc animates images



Aliasing as a feature, not a bug

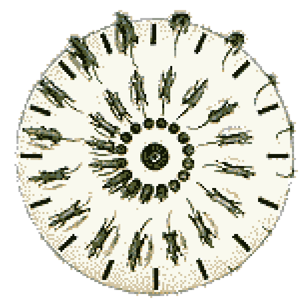
# Phenakistoscope

Invented independently by 2 people in 1832

Disc mounted on spindle

Viewed through slots with images facing mirror

Turning disc animates images



Aliasing as a feature, not a bug

# Zoetrope (1834)

Images arranged on paper band inside a drum

Slits cut in the upper half of the drum

Opposite side viewed as drum rapidly spun

Praxinoscope is a variation on this

Aliasing as a feature, not a bug

# Zoetrope (1834)

Images arranged on paper band inside a drum

Slits cut in the upper half of the drum

Opposite side viewed as drum rapidly spun

Praxinoscope is a variation on this

Aliasing as a feature, not a bug

# Mutoscope (1895)

Coin-operated "flip-book" animation

Picture cards attached to a drum

Popular at sea-side resorts, etc.

# Animation History

First known example of animation:

Humorous phases of funny faces (1906)

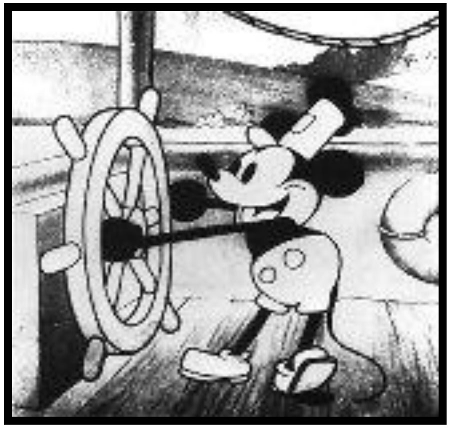
# Key Developments

Plot

Creation of animation studios

Inking on cels

“Felix the Cat”
Otto Messmer (1921)

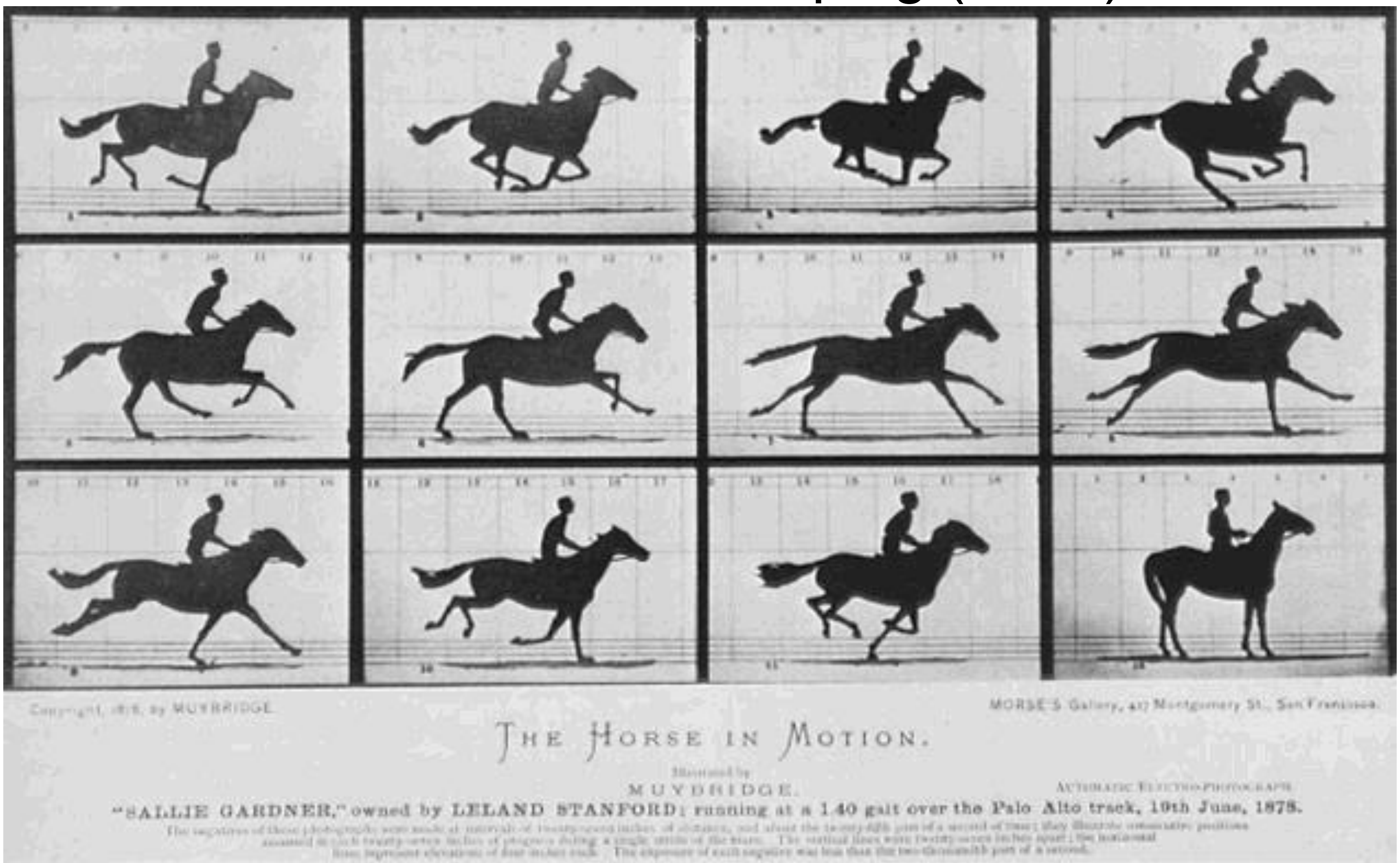


“Steamboat Willie”
Walt Disney (1928)

“Gertie the Dinosaur”
Windsor McCay (1914)

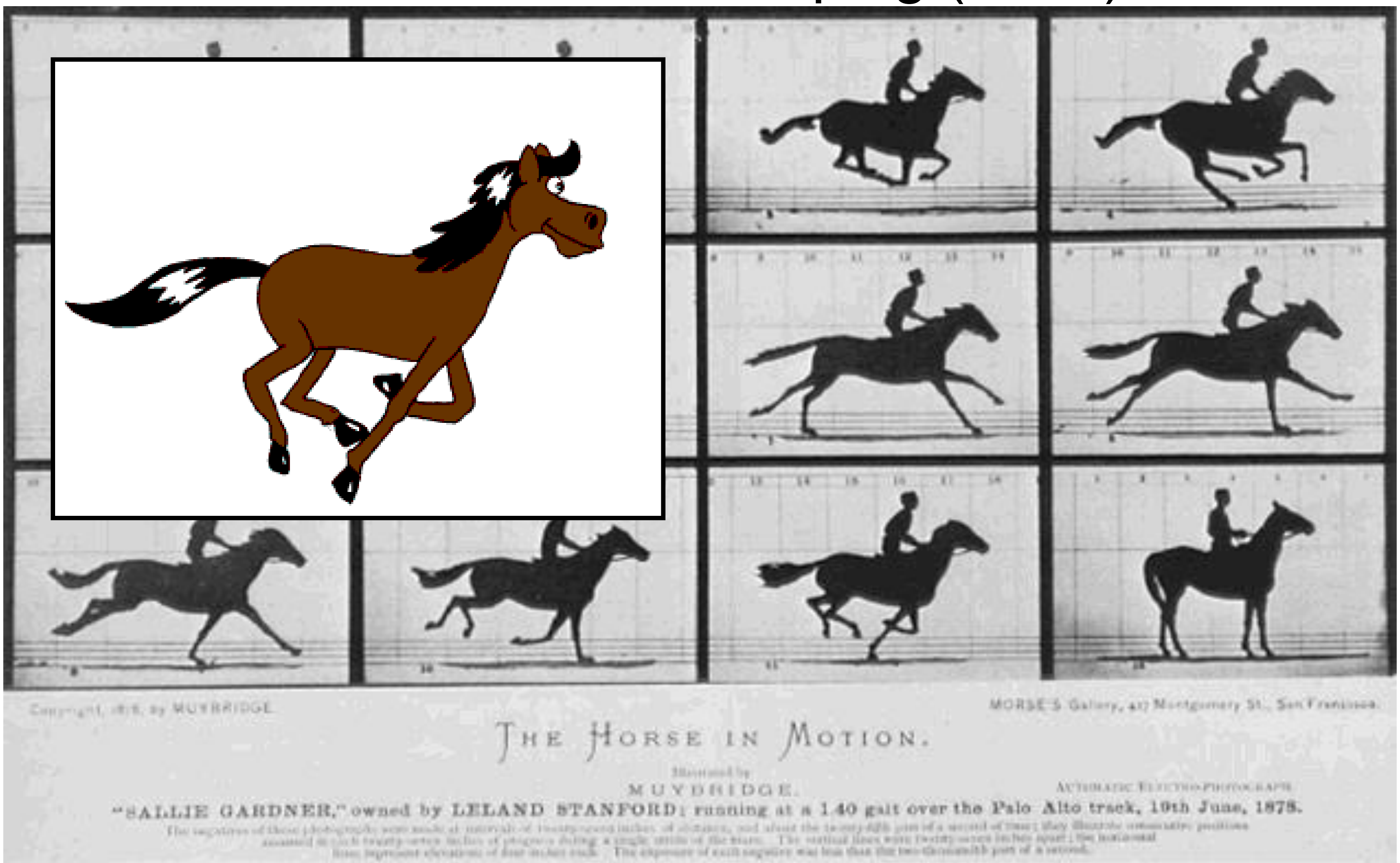
# Key Developments

Max Fleischer invents rotoscoping (1921)



# Key Developments

Max Fleischer invents rotoscoping (1921)



# Key Developments

“Flowers and Trees”, 1932:

- Uses color, wins Academy Award

“Snow White” (aka “Disney’s Folly”), 1937:

- $1.4 million to make
- 750 artists
- Highest grossing ($8 million)


“Flowers and Trees”
Walt Disney

“Snow White”
Walt Disney

# Animation Uses

- Entertainment
- Education
- Propaganda

# Principles of Traditional Animation

How do we communicate aspects of the animation that are not strictly visual?

- Rigidity
- Weight
- Mood
- Intent
- Focus
- Etc.

See, for example, *The Illusion of Life: Disney Animation* for Disney's 12 basic principals of animation.

Luxo Jr.

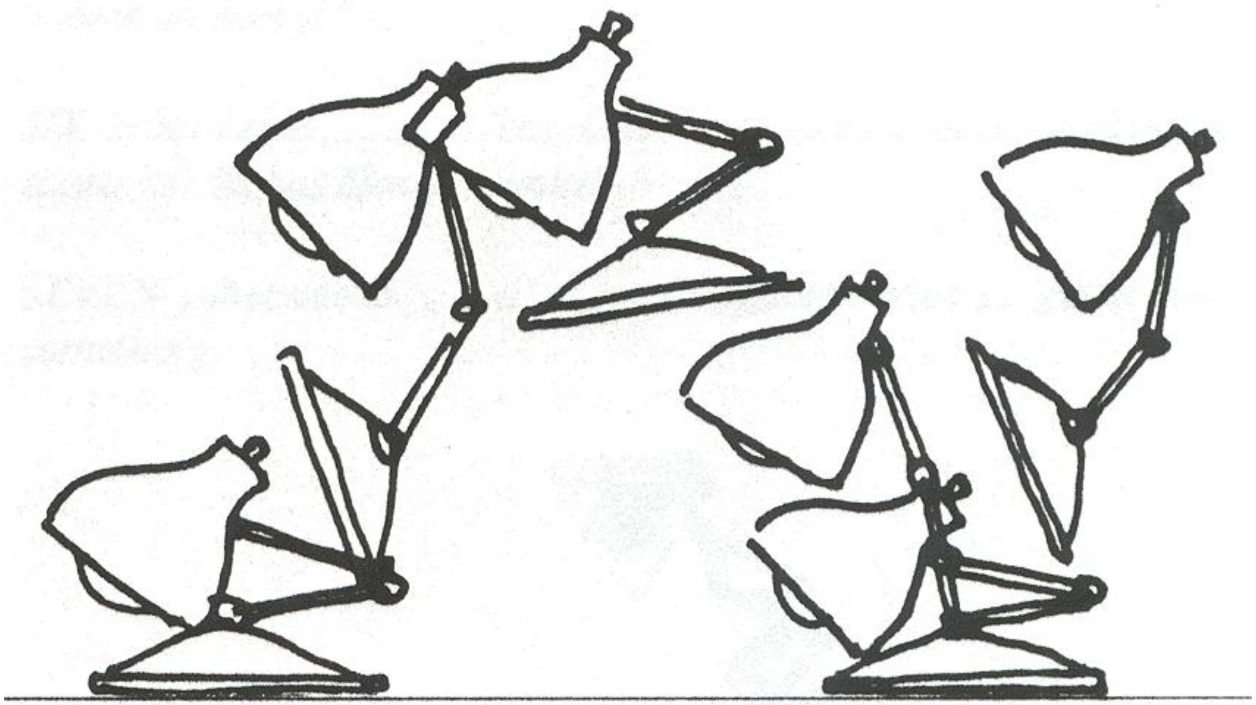
https://en.wikipedia.org/wiki/12_basic_principles_of_animation

# Overview

- Some early animation history
- Computer animation
  - Keyframe animation
  - Articulated figures
  - Kinematics and dynamics

# Keyframe Animation

Define character poses at specific time steps called “keyframes”

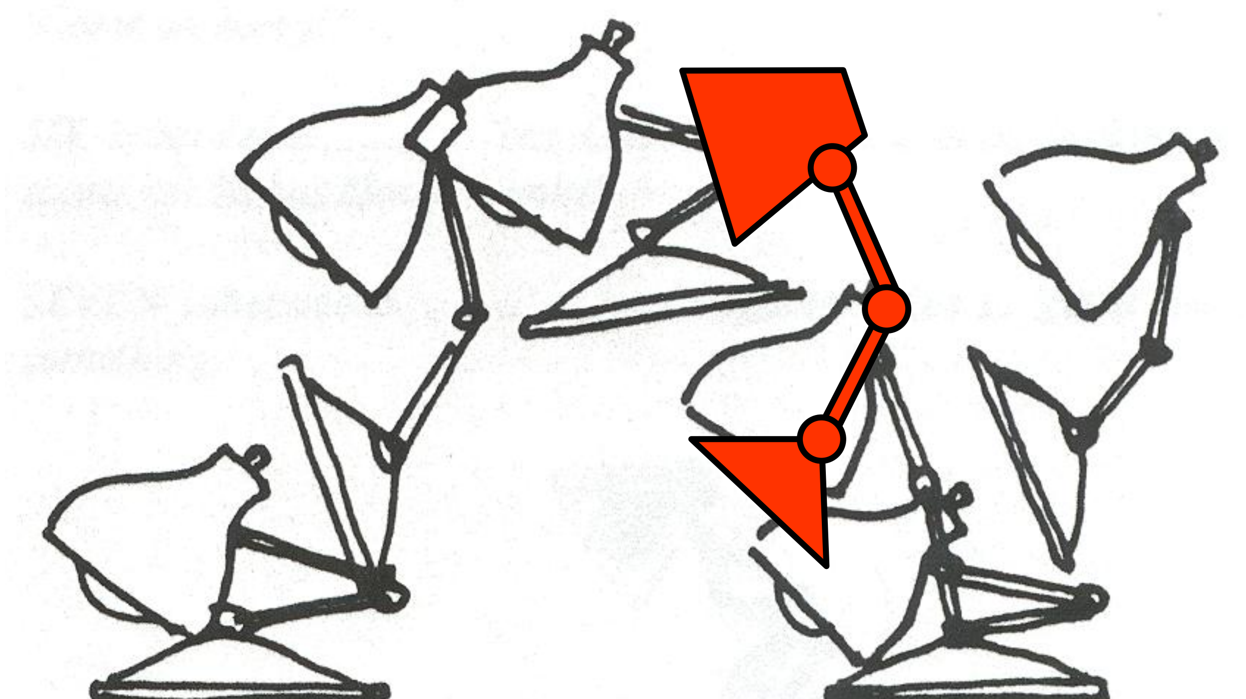


Lasseter `87

# Keyframe Animation

Define character poses at specific time steps called “keyframes”

Interpolate/blend variables describing keyframes to determine poses for character “in-between”

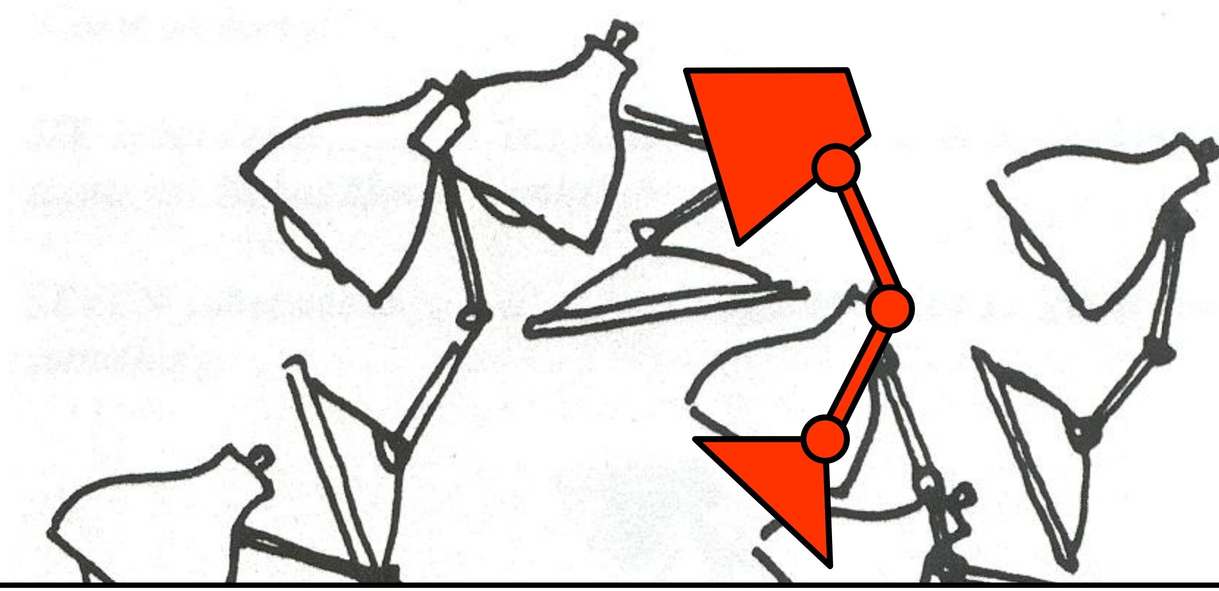


Lasseter `87

# Keyframe Animation

Define character poses at specific time steps called "keyframes"

Interpolate/blend variables describing keyframes to determine poses for character "in-between"

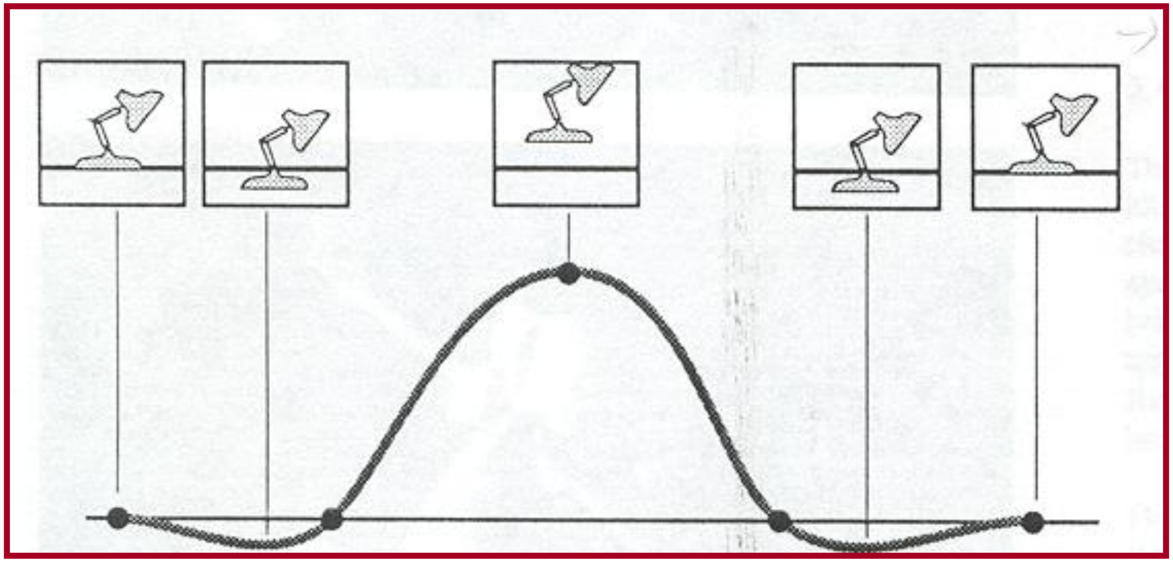


Can use your favorite spline to blend (assuming it satisfies required conditions, smoothness/convex-hull-containment/etc.)

# Keyframe Animation

Define character poses at specific time steps called "keyframes"

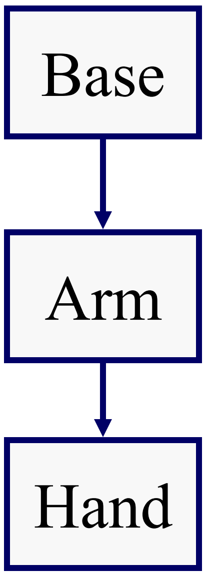
Interpolate/blend variables describing keyframes to determine poses for character "in-between"



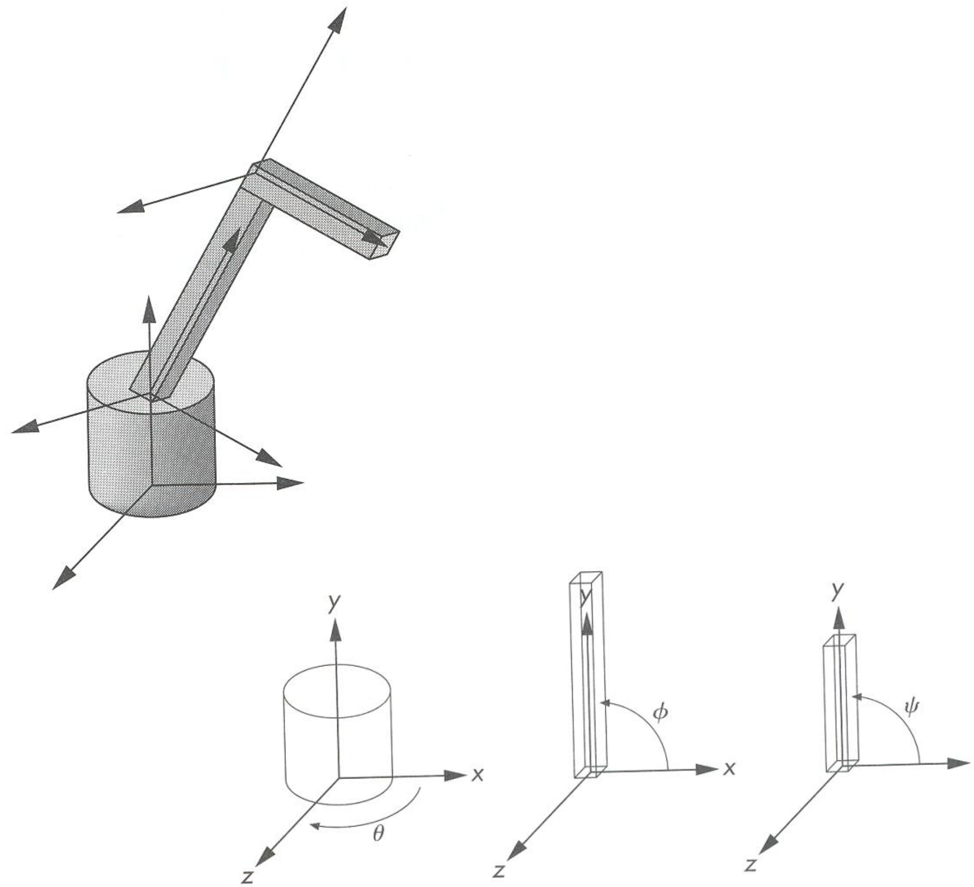
Choose a spline with the properties the application demands (e.g. convex hull containment)

# Articulated Figures

Character poses described by set of rigid bodies connected by “joints”



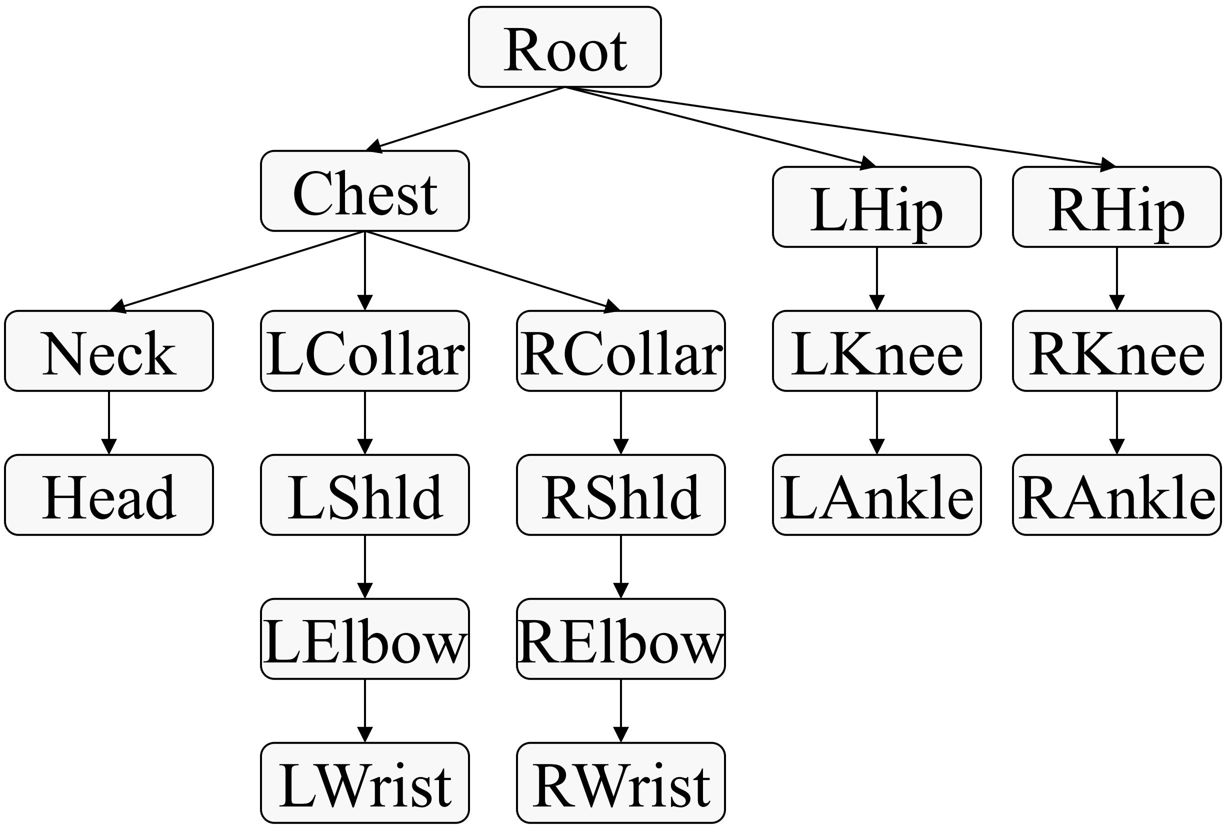

Scene Graph

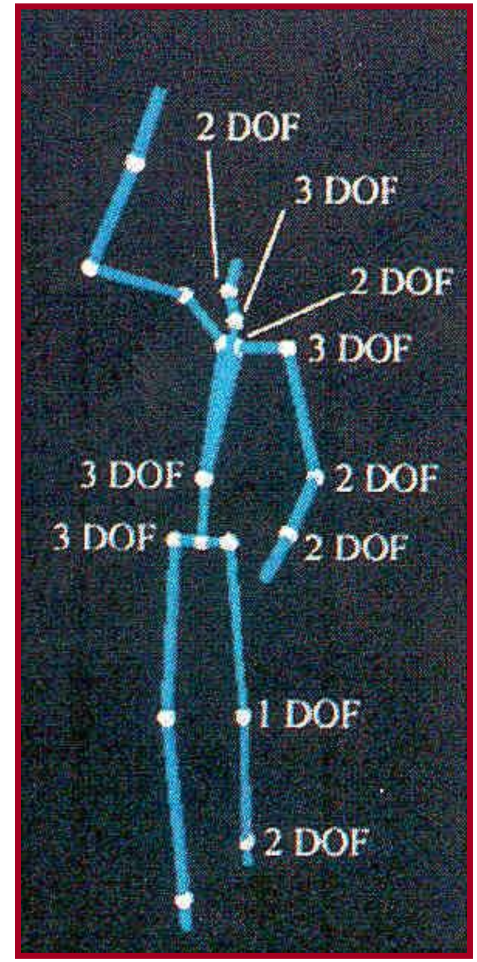



Angel Figures 8.8 & 8.9

# Articulated Figures

Character poses described by set of rigid bodies connected by “joints”

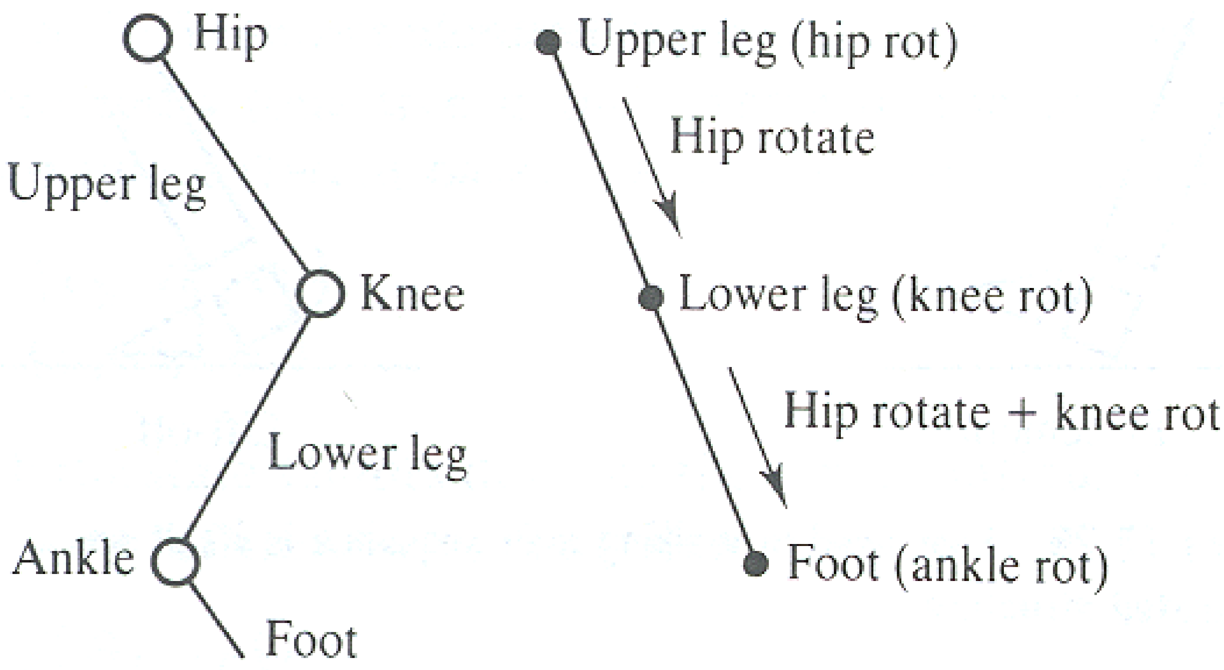



Rose et al. `96

# Example: Walk Cycle

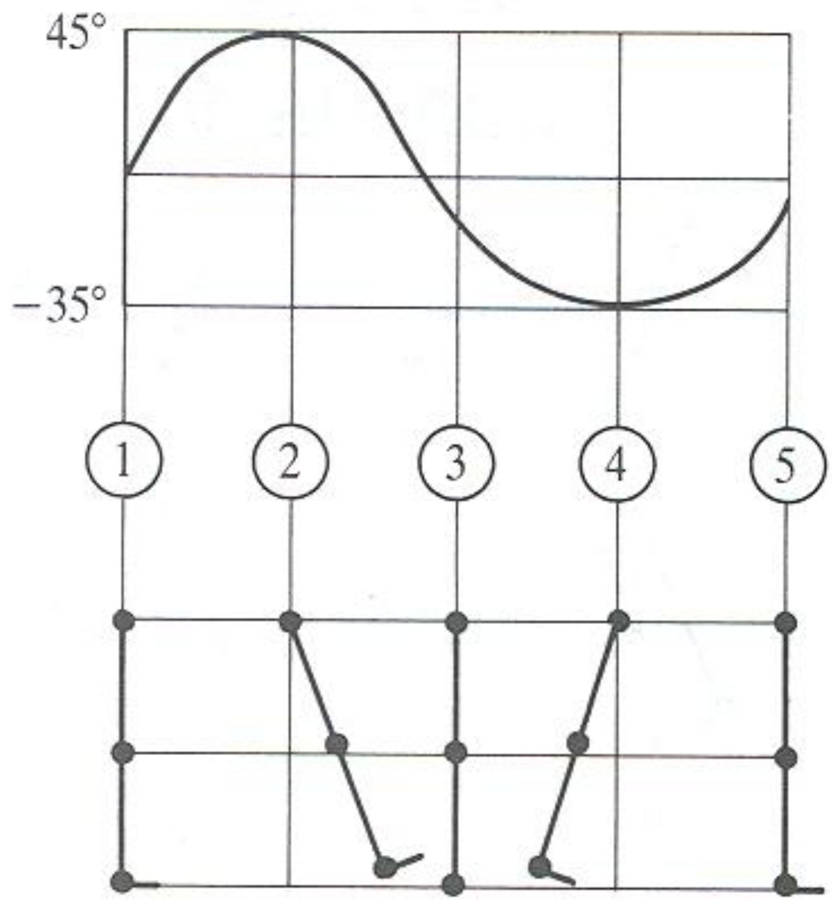
Articulated figure:




Watt & Watt

# Example: Walk Cycle

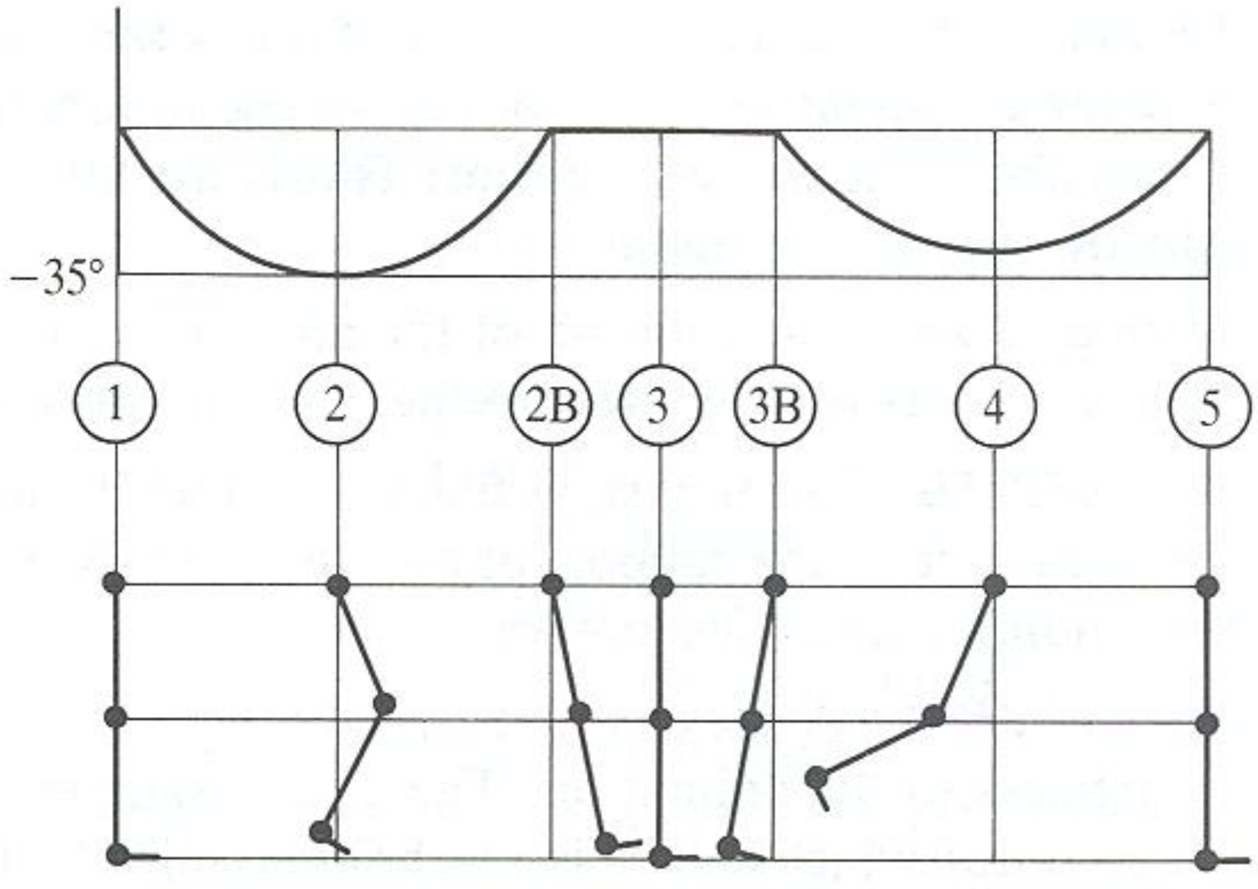
Hip joint orientation:




Watt & Watt

# Example: Walk Cycle

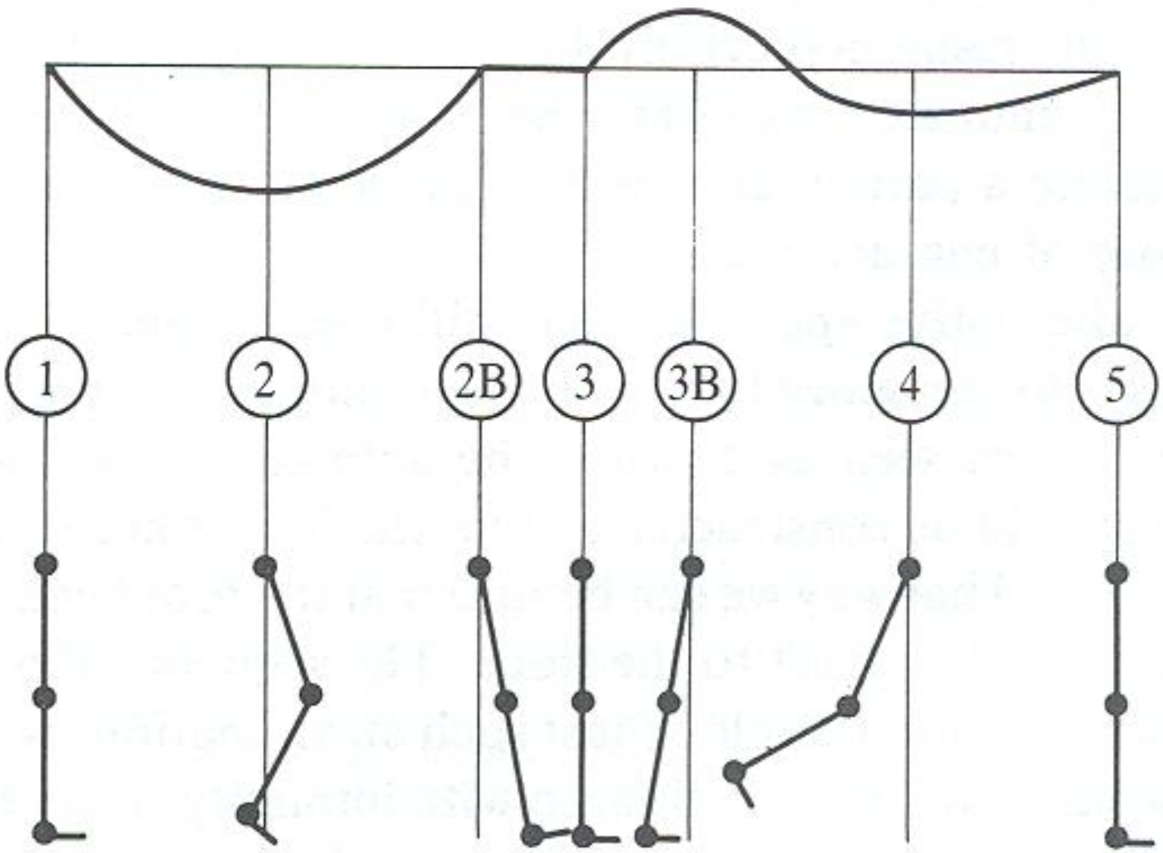
Knee joint orientation:




Watt & Watt

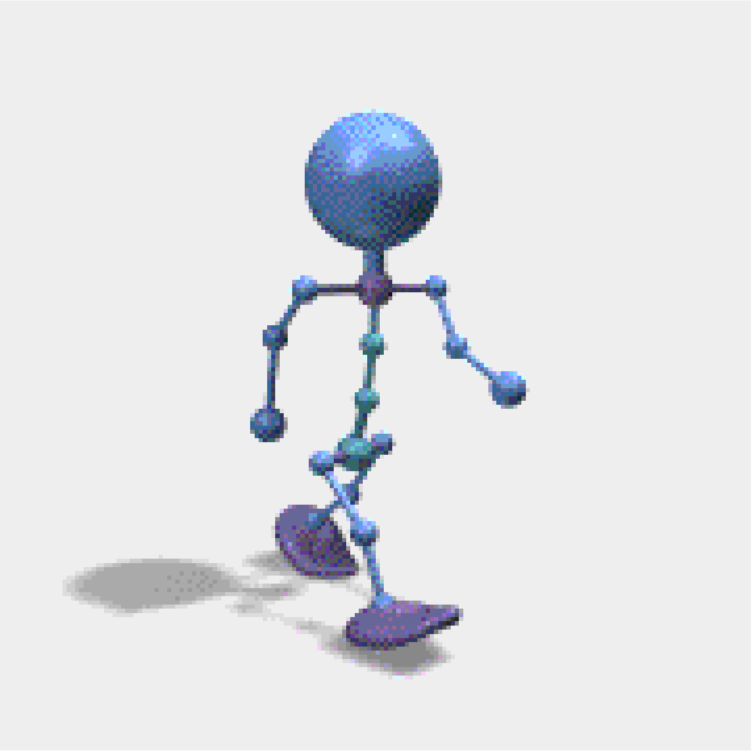
# Example: Walk Cycle

Ankle joint orientation:




Watt & Watt

# Example: Walk Cycle



http://www.ischool.utexas.edu/~luna73/architecture/

Will discuss the how of blending joint parameters next time.

# Keyframe Animation

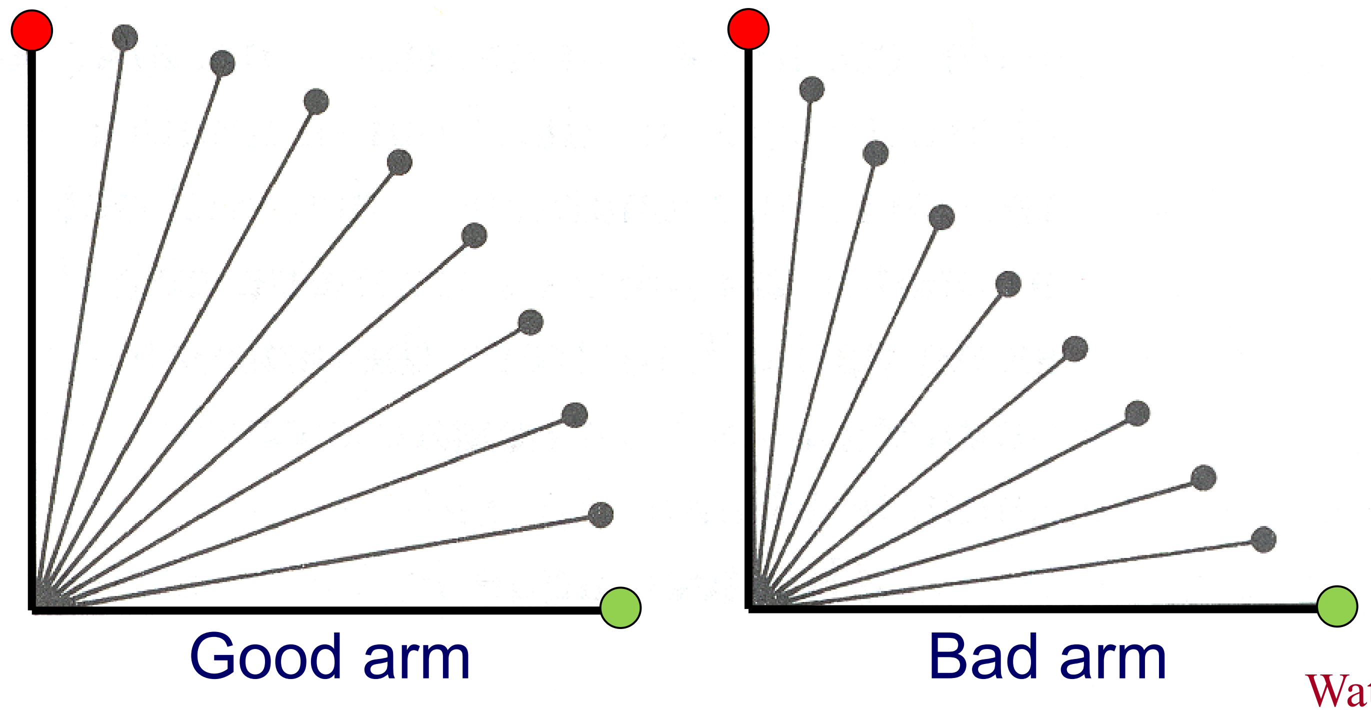
Q: Why interpolate/blend joint parameters rather than vertex positions?

A: For translations, it doesn't make a difference (assuming the blend is translation equivariant).

# Keyframe Animation

Q: Why interpolate/blend joint parameters rather than vertex positions?

A: For rotations, it could lead to geometric distortion.




Watt & Watt

# Kinematics and Dynamics

Kinematics: *Study of motion w/o regard for the cause*

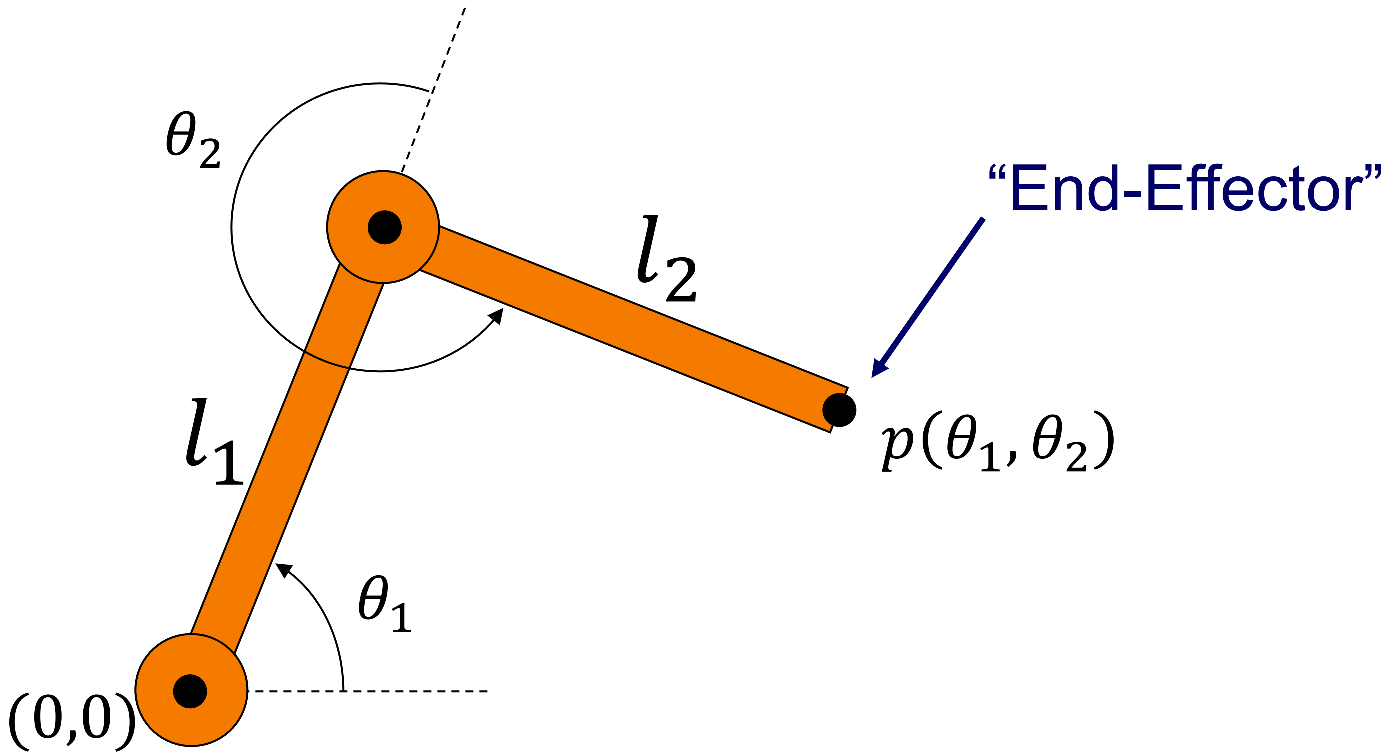
- Considers only motion
- Determined by positions, velocities, accelerations

Dynamics: *Study of the cause of motion*

- Considers underlying forces and interactions
- Compute motion from initial conditions and physics

# Example: 2-Link Structure

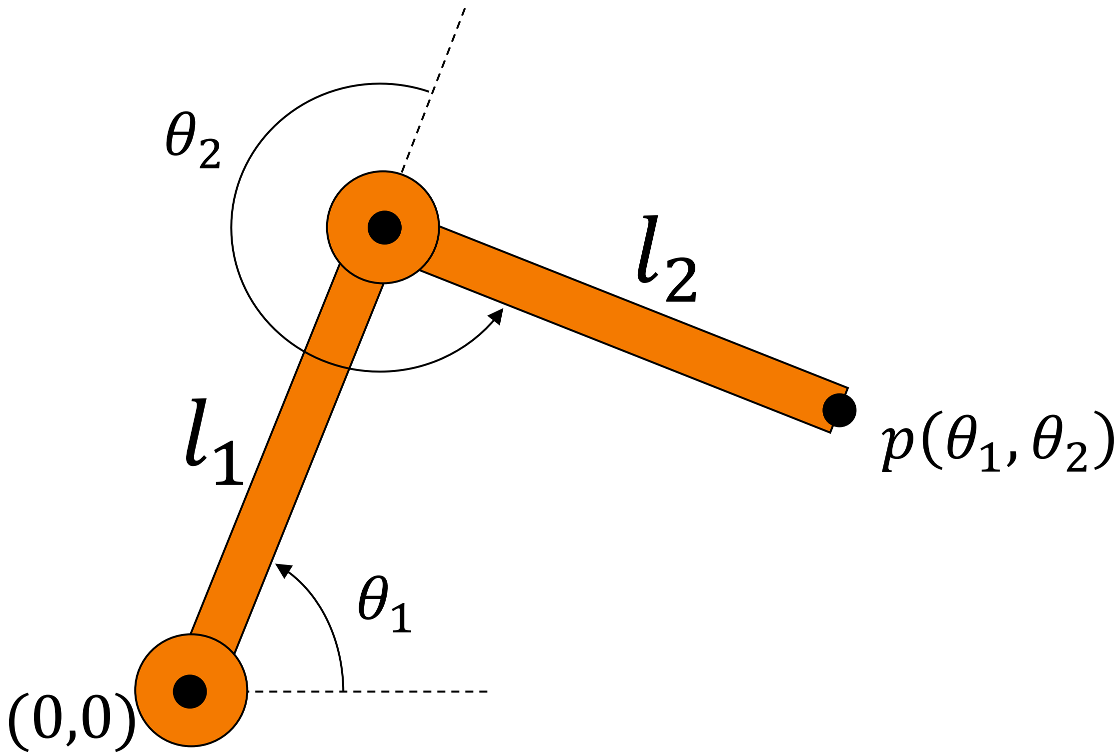
Two links connected by rotational joints



# Forward Kinematics

Animator specifies joint angles: $\theta_1$ and $\theta_2$

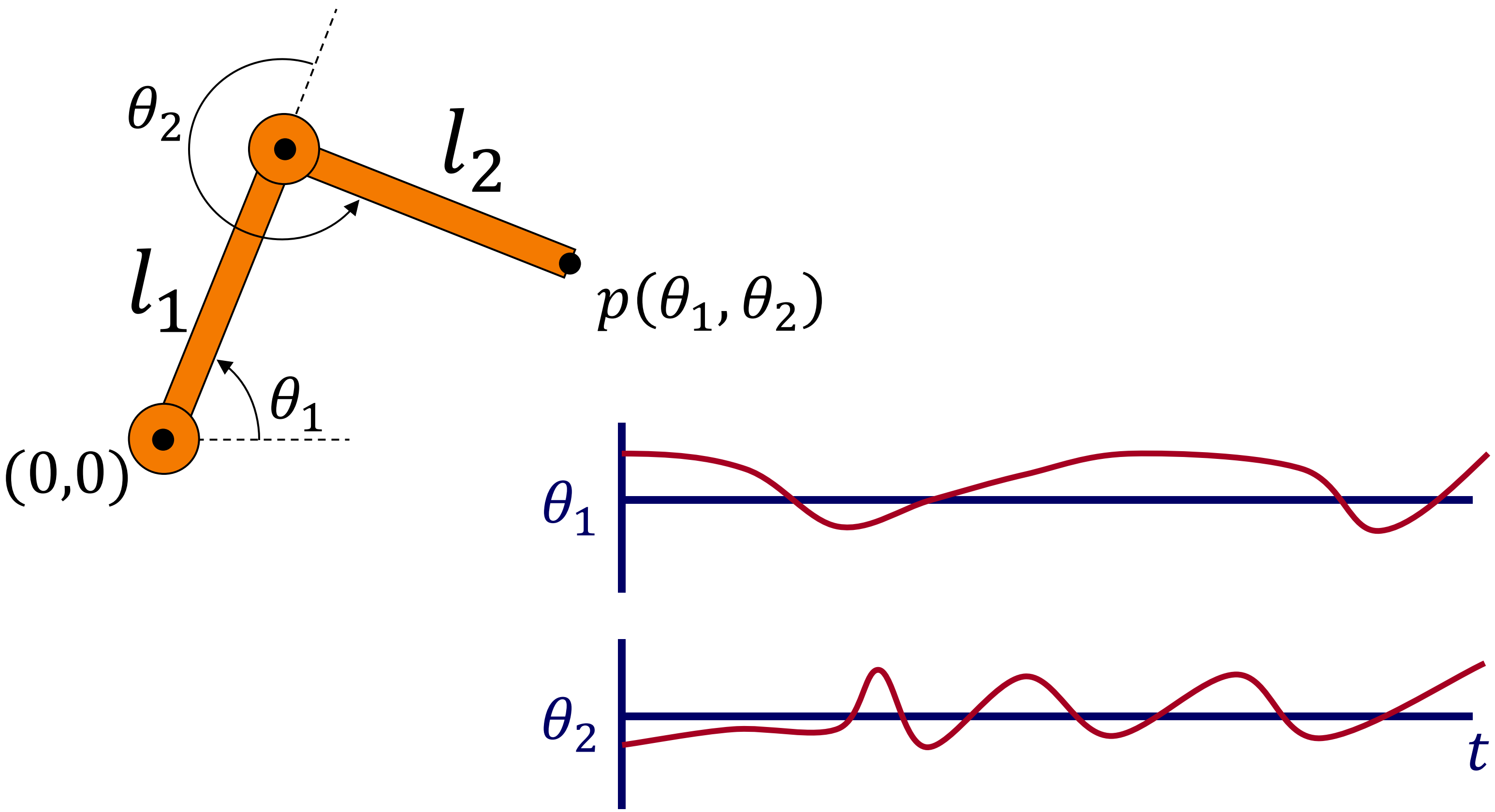
Computer finds positions of end-effector: $p(\theta_1, \theta_2)$




$$p(\theta_1, \theta_2) = (l_1 \cos\theta_1 + l_2(\cos(\theta_1 + \theta_2)), l_1 \sin\theta_1 + l_2 \sin(\theta_1 + \theta_2))$$

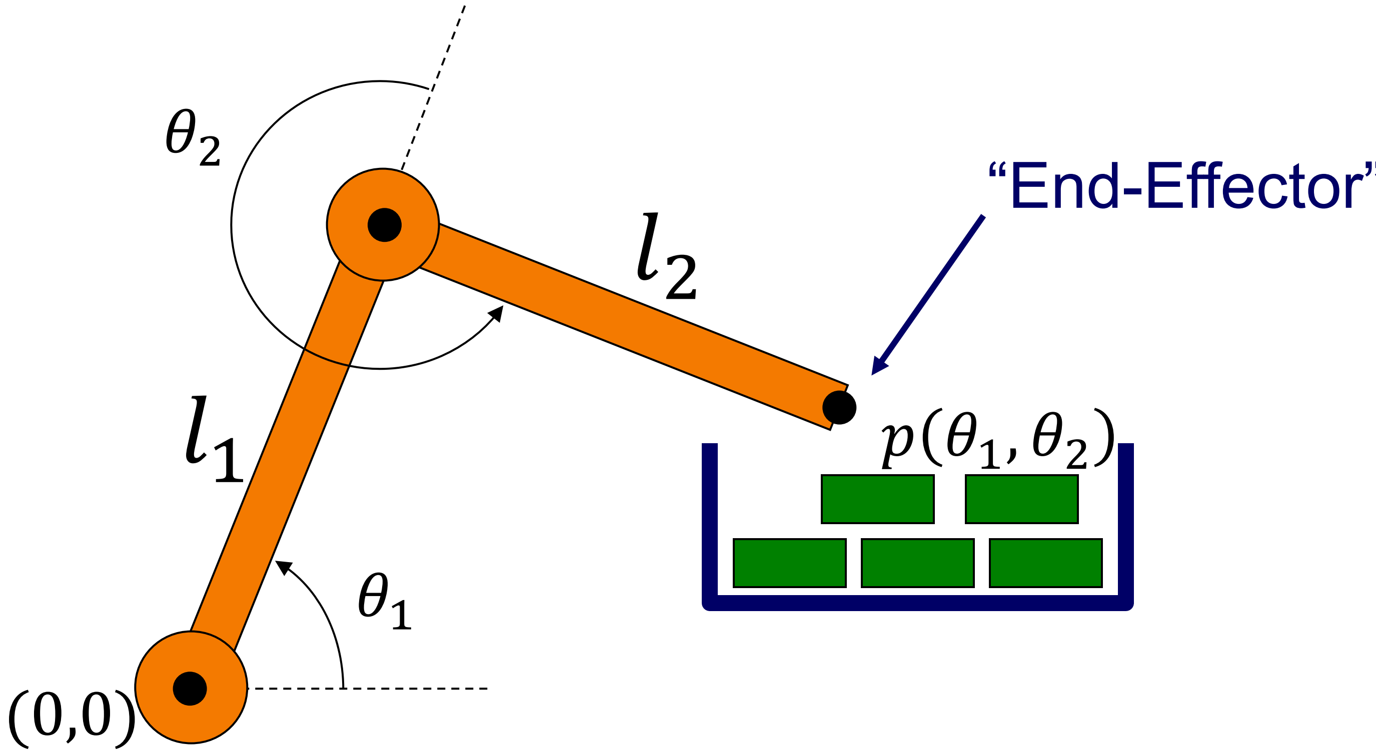
# Forward Kinematics

Joint motions can be specified by spline curves



# Example: 2-Link Structure

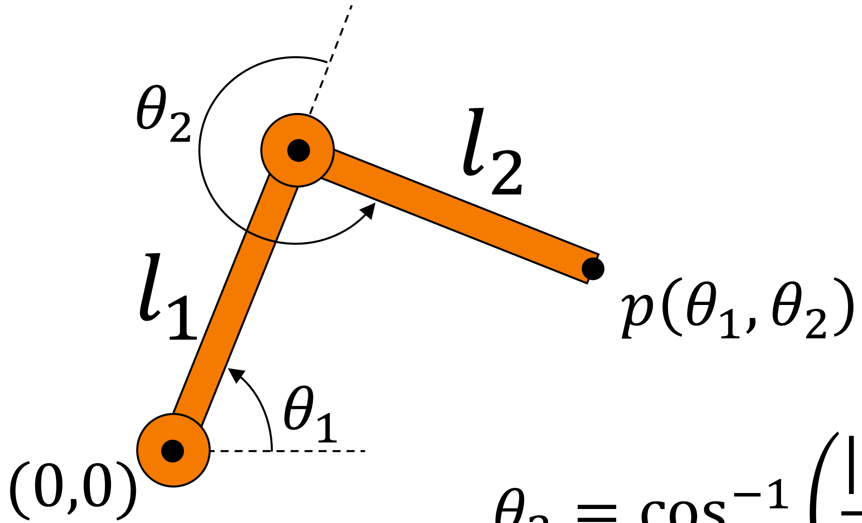
What if animator knows position of “end-effector”



# Inverse Kinematics

Animator specifies end-effector positions: $p$

Computer finds joint angles: $\theta_1$ and $\theta_2$

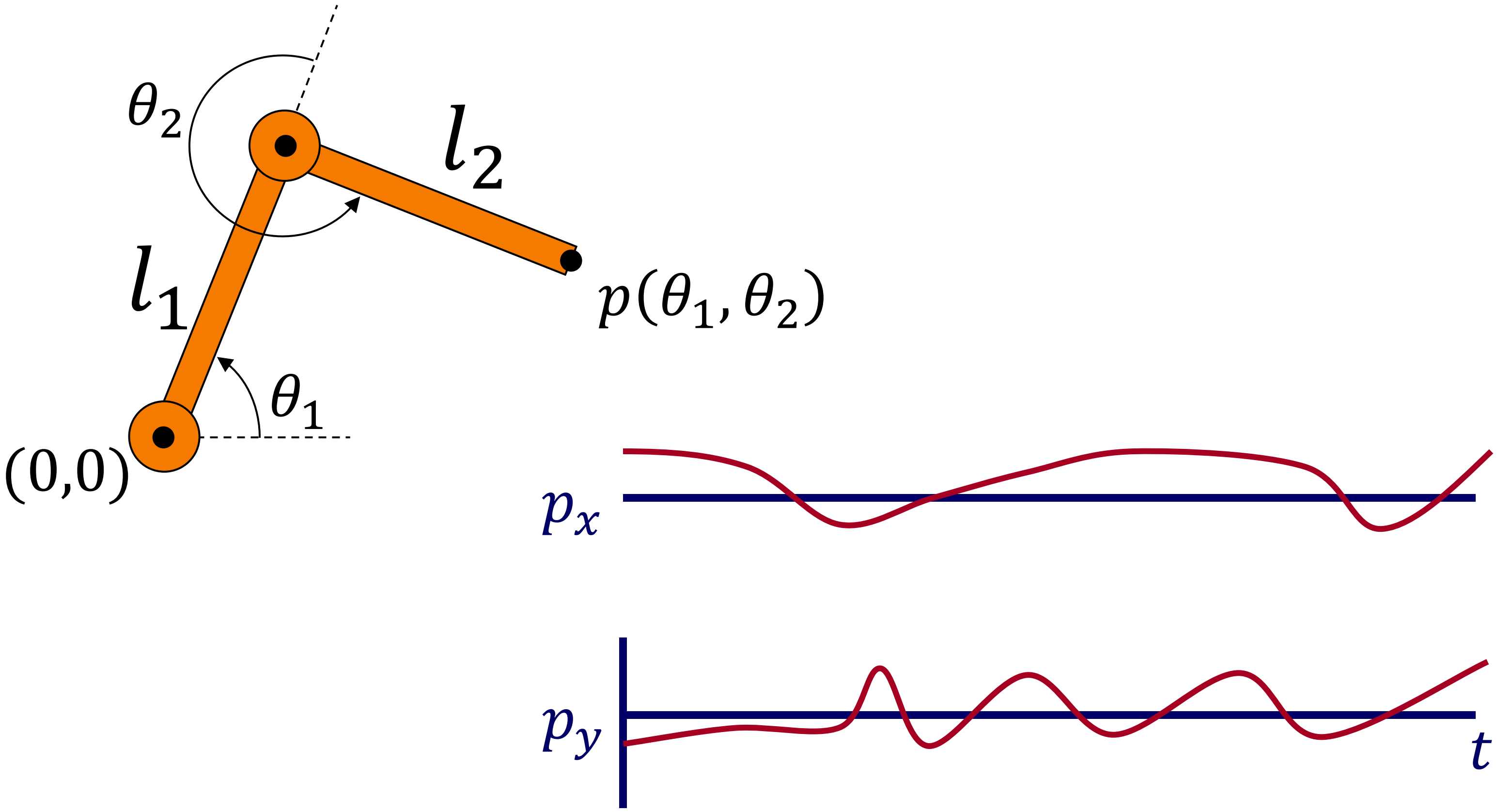



$$\theta_2 = \cos^{-1}\left(\frac{|p|^2 - l_1^2 - l_2^2}{2 l_1 l_2}\right)$$

$$\theta_1 = \tan^{-1}\left(\frac{p_y}{p_x}\right) - \tan^{-1}\left(\frac{l_2 \sin\theta_2}{l_1 + l_2 \cos\theta_2}\right)$$

# Inverse Kinematics

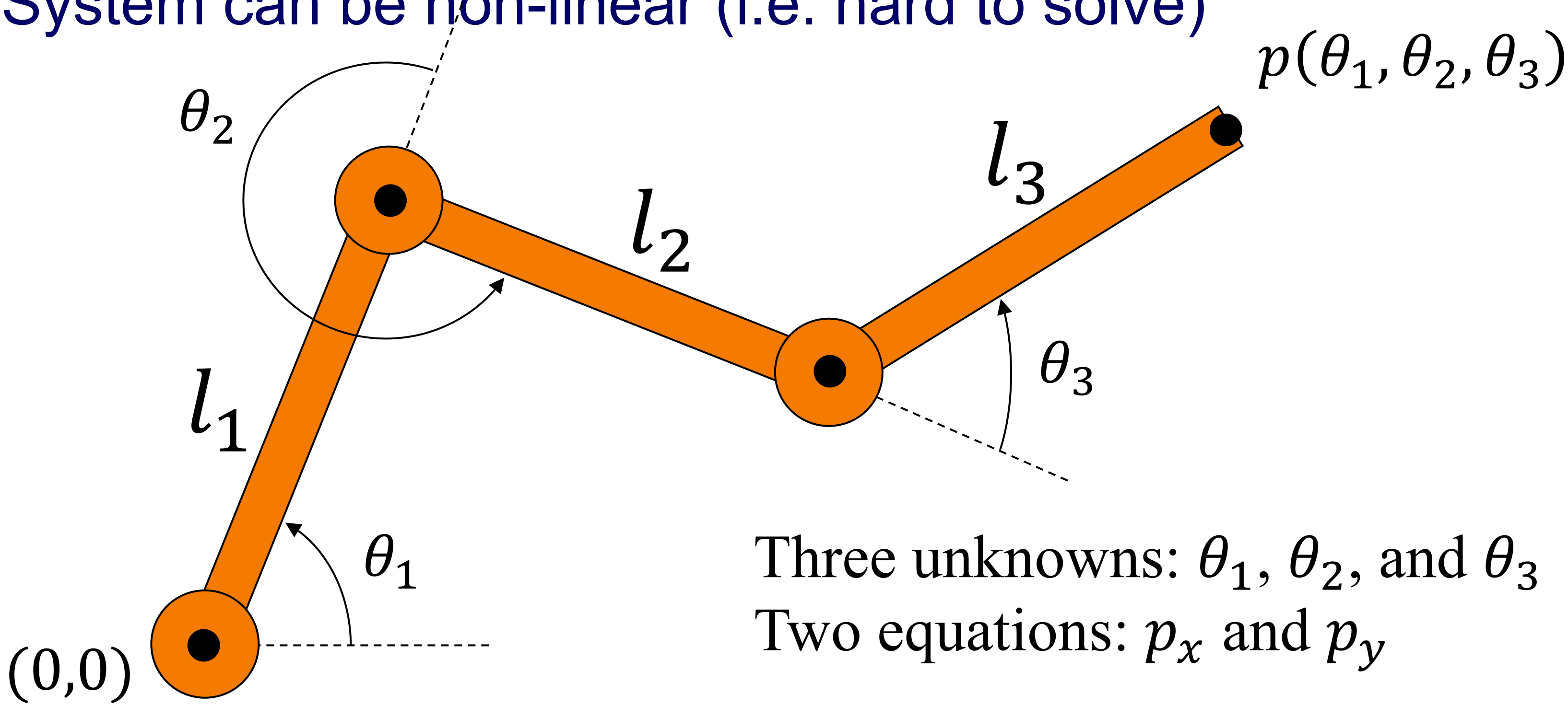
End-effector positions can be specified by splines



# Inverse Kinematics

Challenges for more complex structures

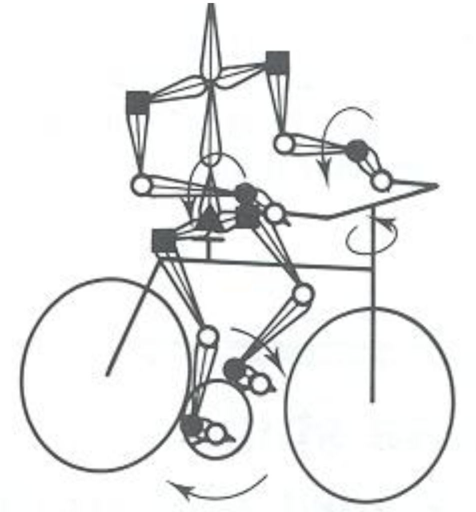
- May have multiple solutions solution
  - » May be able to find best/closest solution
- May not have any solutions
- System can be non-linear (i.e. hard to solve)




Three unknowns: $\theta_1$, $\theta_2$, and $\theta_3$
Two equations: $p_x$ and $p_y$

# Summary of Kinematics

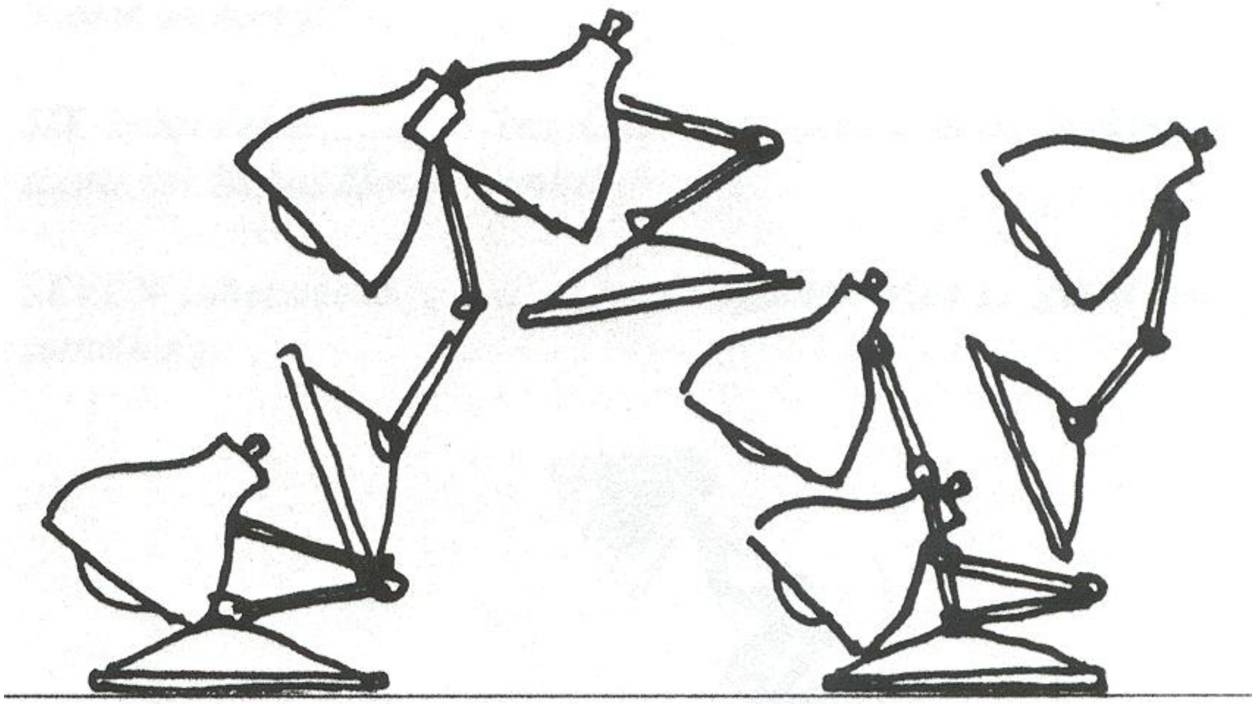
- Forward kinematics
  - Specify conditions (joint angles)
  - Compute positions of end-effectors
- Inverse kinematics
  - “Goal-directed” motion
  - Specify goal positions of end effectors
  - Compute conditions required to achieve goals



Inverse kinematics provides easier specification for many animation tasks, but it is computationally more difficult

# Dynamics

Simulate physics to obtain motion that is responsive/realistic

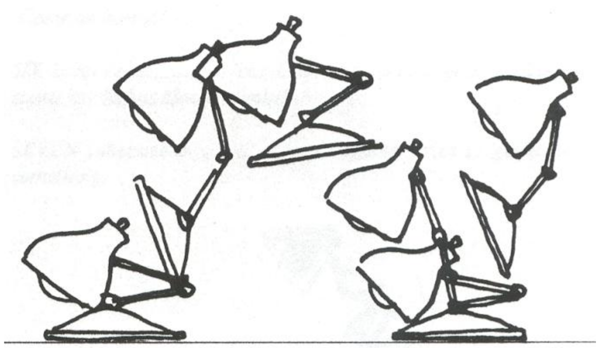


Lasseter `87

# Dynamics

Animator specifies constraints:

- What the character's physical structure is
  - e.g., articulated figure
- What the character has to do
  - e.g., jump from here to there within specific time
- What other physical structures are present
  - e.g., floor to push off and land
- How the motion should be performed
  - e.g., minimize energy



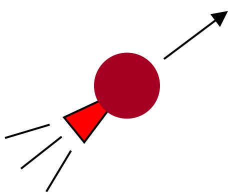
# Dynamics

Computer finds the "best" physical motion satisfying the constraints (e.g. by minimizing some objective)

Example: particle with jet propulsion

- $\mathbf{x}(t)$ is position of particle at time $t$
- $\mathbf{f}(t)$ is the directional force of jet propulsion at time $t$
- Particle's equation of motion is:

$$0 = m(\ddot{\mathbf{x}} - \mathbf{g}) - \mathbf{f}$$



$$\dot{\mathbf{x}} = \frac{\partial \mathbf{x}}{\partial t} \qquad \ddot{\mathbf{x}} = \frac{\partial^2 \mathbf{x}}{\partial t^2}$$

- In a unit of time, move from $\mathbf{a}$ to $\mathbf{b}$ minimizing

$$E(\mathbf{x}) = \int_0^1 \|\mathbf{f}(t)\|^2 dt = \int_0^1 \|m(\ddot{\mathbf{x}} - \mathbf{g})(t)\|^2 dt$$

  Such that:

$$\mathbf{x}(0) = \mathbf{a},\ \dot{\mathbf{x}}(0) = \mathbf{0},\ \mathbf{x}(1) = \mathbf{b}, \text{ and } \dot{\mathbf{x}}(1) = \mathbf{0}$$

# Dynamics

- Discretize time into $N$ uniform intervals (of size $h = \frac{1}{N}$)
- Solve for discrete positions $\{\mathbf{x}_0, \cdots, \mathbf{x}_N\}$, w/ $\mathbf{x}_i = \mathbf{x}(i \cdot h)$
- Compute discrete velocities:
$$\dot{\mathbf{x}}_i = \frac{\mathbf{x}_i - \mathbf{x}_{i-1}}{h}$$
- Compute discrete acceleration:
$$\ddot{\mathbf{x}}_i = \frac{\dot{\mathbf{x}}_{i+1} - \dot{\mathbf{x}}_i}{h} = \frac{\mathbf{x}_{i+1} - 2\mathbf{x}_i + \mathbf{x}_{i-1}}{h^2}$$
- Compute discrete force:
$$\mathbf{f}_i = m(\ddot{\mathbf{x}}_i - \mathbf{g}) = m\left(\frac{\mathbf{x}_{i+1} - 2\mathbf{x}_i + \mathbf{x}_{i-1}}{h^2} - \mathbf{g}\right)$$

# Dynamics

$$\mathbf{f}_i = m\left(\frac{\mathbf{x}_{i+1} - 2\mathbf{x}_i + \mathbf{x}_{i-1}}{h^2} - \mathbf{g}\right)$$

---

Optimization ⇒ minimizing a quadratic energy:

$$E(\mathbf{x}) = \int_0^1 \|m(\ddot{\mathbf{x}} - \mathbf{g})\|^2 dt$$

$$\Downarrow$$

$$E(\{\mathbf{x}_0, \ldots, \mathbf{x}_n\}) = \sum_i \|m(\ddot{\mathbf{x}}_i - \mathbf{g})\|^2$$

$$= hm^2 \sum_i \left\|\frac{\mathbf{x}_{i+1} - 2\mathbf{x}_i + \mathbf{x}_{i-1}}{h^2} - \mathbf{g}\right\|^2$$

subject to $\mathbf{x}_{-1} = \mathbf{x}_0 = \mathbf{a}$ and $\mathbf{x}_N = \mathbf{x}_{N+1} = \mathbf{b}$.

# Dynamics

$$\mathbf{f}_i = m\left(\frac{\mathbf{x}_{i+1} - 2\mathbf{x}_i + \mathbf{x}_{i-1}}{h^2} - \mathbf{g}\right)$$

---

Optimization ⇒ minimizing a quadratic energy:

$$E(\mathbf{x}) = \int_0^1 \|m(\ddot{\mathbf{x}} - \mathbf{g})\|^2 dt$$

$$\Downarrow$$

$$E(\{\mathbf{x}_0, \dots, \mathbf{x}_n\}) = \sum_i \|m(\ddot{\mathbf{x}}_i - \mathbf{g})\|^2$$

$$= hm^2 \sum \left\|\frac{\mathbf{x}_{i+1} - 2\mathbf{x}_i + \mathbf{x}_{i-1}}{\ }- \mathbf{g}\right\|^2$$

To minimize, need to find where the gradient is zero. Since the energy is quadratic, the gradient is linear.

Reduces to solving a *linear* system of equations.

# Dynamics

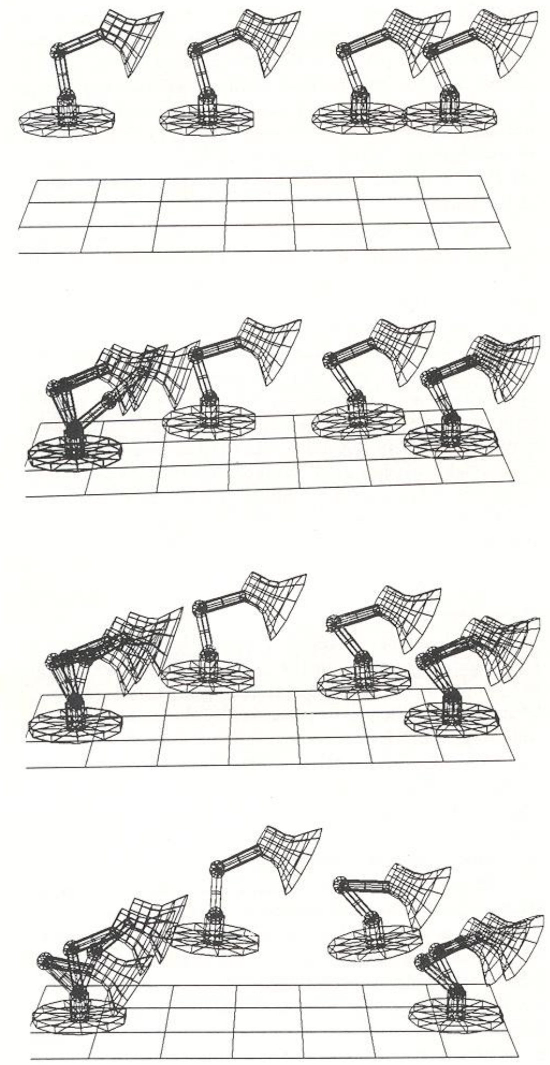
For simple scenarios:

Solve a linear system

$$\mathbf{Ax} = \mathbf{b}$$

For complex scenarios:

Solve using iterative optimization techniques

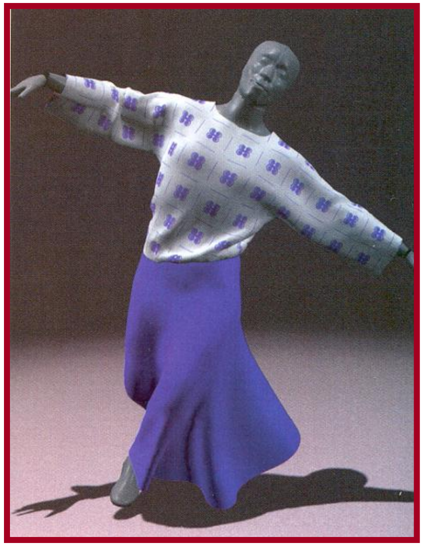


Witkin & Kass `88

# Dynamics

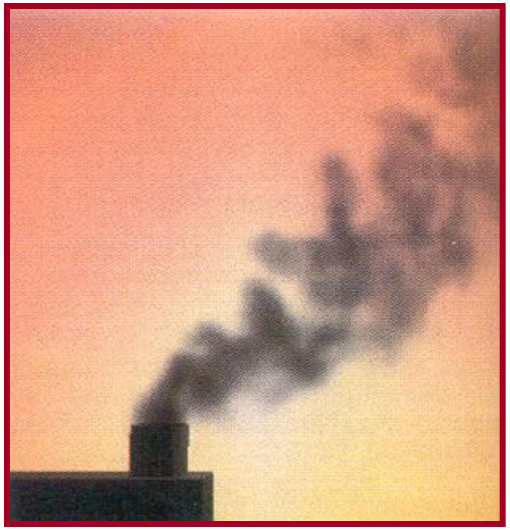
Other physical simulations:

- Rigid bodies
- Soft bodies
- Cloth
- Liquids
- Gases
- etc.



Cloth

*(Baraff & Witkin `98)*



Hot Gases

*(Foster & Metaxas `97)*

Demo

# Dynamics

Advantages:

- Free animator from having to specify details of physically realistic motion with spline curves
- Easy to vary motions due to new parameters and/or new constraints

Challenges:

- Specifying constraints and objective functions
- Avoiding local minima during optimization