# Kinematics

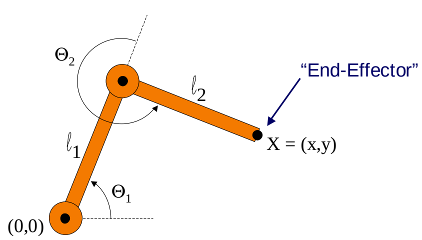
Thomas Funkhouser
Princeton University
COS 426, Spring 2004

## Overview

- Kinematics
  - Considers only motion
  - Determined by positions, velocities, accelerations
- Dynamics
  - Considers underlying forces
  - Compute motion from initial conditions and physics

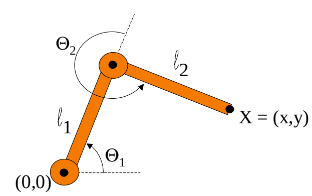
## Example: 2-Link Structure

- Two links connected by rotational joints




## Forward Kinematics

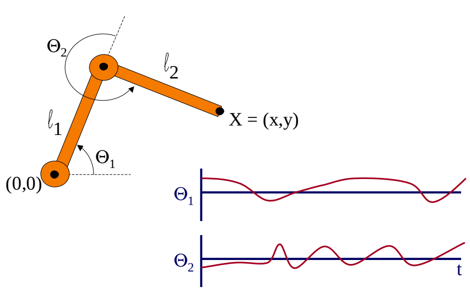
- Animator specifies joint angles: $\Theta_1$ and $\Theta_2$
- Computer finds positions of end-effector: X




$$X = (l_1 \cos\Theta_1 + l_2 \cos(\Theta_1 + \Theta_2), l_1 \sin\Theta_1 + l_2 \sin(\Theta_1 + \Theta_2))$$

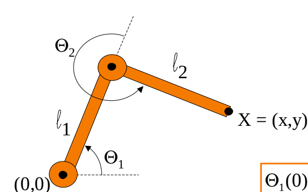
## Forward Kinematics

- Joint motions can be specified by spline curves




## Forward Kinematics

- Joint motions can be specified by initial conditions and velocities

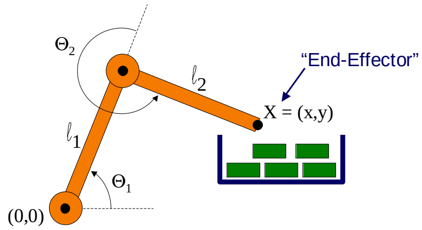



$$\Theta_1(0) = 60^\circ \qquad \Theta_2(0) = 250^\circ$$

$$\frac{d\Theta_1}{dt} = 1.2 \qquad \frac{d\Theta_2}{dt} = -0.1$$

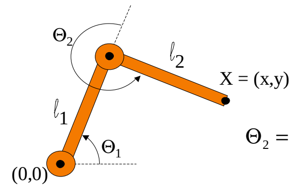
## Example: 2-Link Structure

- What if animator knows position of "end-effector"




## Inverse Kinematics

- Animator specifies end-effector positions: X
- Computer finds joint angles: $\Theta_1$ and $\Theta_2$:

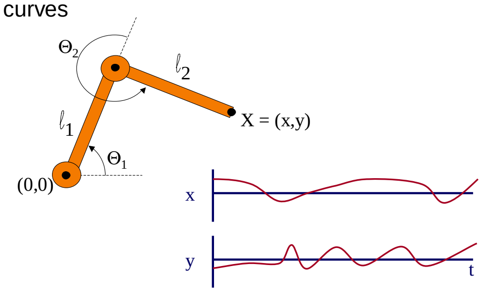



$$\Theta_2 = \cos^{-1}\left(\frac{x^2 + x^2 - l_1^2 - l_2^2}{2 l_1 l_2}\right)$$

$$\Theta_1 = \frac{-(l_2 \sin(\Theta_2))x + (l_1 + l_2 \cos(\Theta_2))y}{(l_2 \sin(\Theta_2))y + (l_1 + l_2 \cos(\Theta_2))x}$$

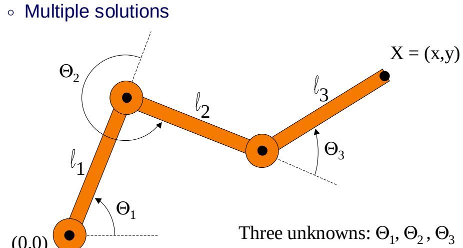
## Inverse Kinematics

- End-effector postions can be specified by spline curves




## Inverse Kinematics

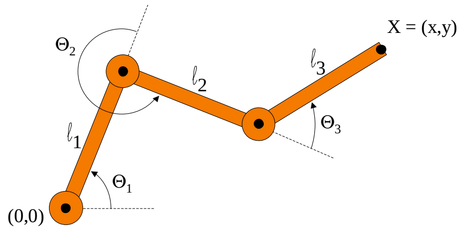
- Problem for more complex structures
  - System of equations is usually under-defined
  - Multiple solutions




Three unknowns: $\Theta_1$, $\Theta_2$, $\Theta_3$
Two equations: x, y

## Inverse Kinematics

- Solution for more complex structures:
  - Find best solution (e.g., minimize energy in motion)
  - Non-linear optimization



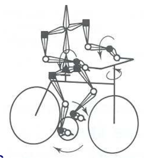

## Inverse Kinematics

"Ballboy"

Fujito, Milliron, Ngan, & Sanocki
Princeton University

## 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



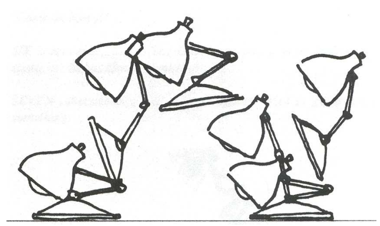
Inverse kinematics provides easier specification for many animation tasks, but it is computationally more difficult

## Overview

- Kinematics
  - Considers only motion
  - Determined by positions, velocities, accelerations
- Dynamics
  - Considers underlying forces
  - Compute motion from initial conditions and physics

## Dynamics

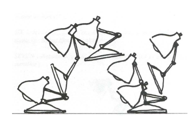
- Simulation of physics insures realism of motion



Lasseter `87

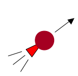
## Spacetime Constraints

- 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 time t
  - What other physical structures are present
    - » e.g., floor to push off and land
  - How the motion should be performed
    - » e.g., minimize energy



## Spacetime Constraints

- Computer finds the "best" physical motion satisfying constraints
- Example: particle with jet propulsion
  - **x**(t) is position of particle at time t
  - **f**(t) is force of jet propulsion at time t
  - Particle's equation of motion is:



$$mx'' - f - mg = 0$$

  - Suppose we want to move from a to b within $t_0$ to $t_1$ with minimum jet fuel:

$$\text{Minimize } \int_{t_0}^{t_1} |f(t)^2| dt \quad \text{subject to } x(t_0)=a \text{ and } x(t_1)=b$$

Witkin & Kass `88

## Spacetime Constraints

- Discretize time steps:

$$x'_i = \frac{x_i - x_{i-1}}{h}$$

$$x''_i = \frac{x_{i+1} - 2x_i + x_{i-1}}{h^2}$$

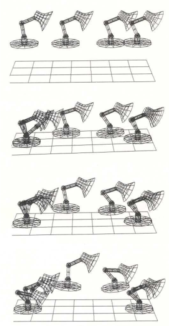
$$m\left(x''_i = \frac{x_{i+1} - 2x_i + x_{i-1}}{h^2}\right) - f_i - mg = 0$$

$$\text{Minimize } h\sum_i |f_i|^2 \quad \text{subject to } x_0=a \text{ and } x_1=b$$

Witkin & Kass `88

## Spacetime Constraints

- Solve with iterative optimization methods



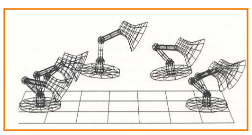
Witkin & Kass `88

## Spacetime Constraints

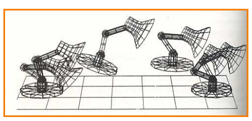
- Advantages:
  - Free animator from having to specify details of physically realistic motion with spline curves
  - Easy to vary motions due to new parameters and/or new constraints
- Challenges:
  - Specifying constraints and objective functions
  - Avoiding local minima during optimization

## Spacetime Constraints

- Adapting motion:



Original Jump

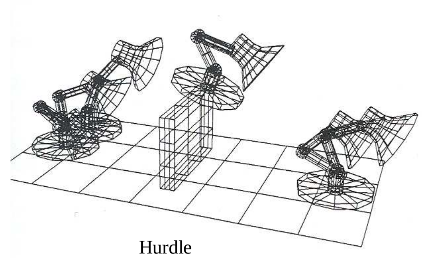


Heavier Base

Witkin & Kass `88

## Spacetime Constraints

- Adapting motion:

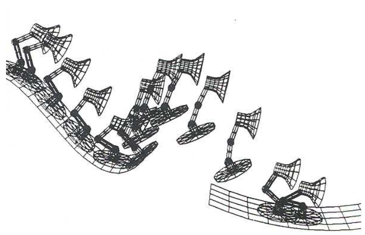


Hurdle

Witkin & Kass `88

## Spacetime Constraints

- Adapting motion:



Ski Jump

Witkin & Kass `88

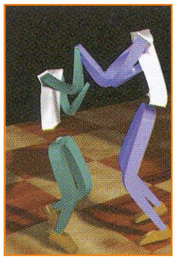
## Spacetime Constraints

- Editing motion:

Li et al. `99

## Spacetime Constraints

- Morphing motion:



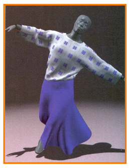
Gleicher `98

## Spacetime Constraints

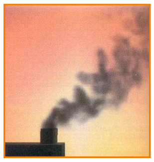
- Advantages:
  - Free animator from having to specify details of physically realistic motion with spline curves
  - Easy to vary motions due to new parameters and/or new constraints
- Challenges:
  - Specifying constraints and objective functions
  - Avoiding local minima during optimization

## Dynamics

- Other physical simulations:
  - Rigid bodies
  - Soft bodies
  - Cloth
  - Liquids
  - Gases
  - etc.



Cloth
*(Baraff & Witkin `98)*



Hot Gases
*(Foster & Metaxas `97)*

## Summary

- Kinematics
  - Forward kinematics
    - » Animator specifies joints (hard)
    - » Compute end-effectors (easy - assn 4!)
  - Inverse kinematics
    - » Animator specifies end-effectors (easier)
    - » Solve for joints (harder)
- Dynamics
  - Space-time constraints
    - » Animator specifies structures & constraints (easiest)
    - » Solve for motion (hardest)
  - Also other physical simulations