# Realistic Character Animation

# Reading

- Jessica Hodgins, et.al, *Animating Human Athletics*, SIGGRAPH '95
- Zoran Popović, *Changing Physics for Character Animation*, SIGGRAPH '00



# Modeling Realistic Motion

- Model muscles
- Environment forces
- Energy consumption
- Individual style



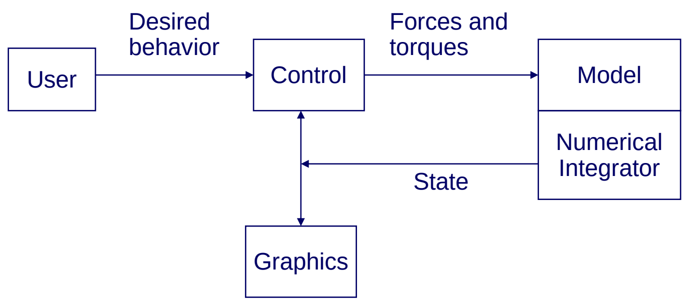
# Two Approaches

- Simulate robot controllers
- Solve a large optimization that obeys laws of physics and minimized energy consumption

# Robot Controllers in Animation



# Control Systems






# Where do the control laws come from?

- Observation
- Biomechanical literature
- Optimization
- Intuition



# Hierarchy of control laws

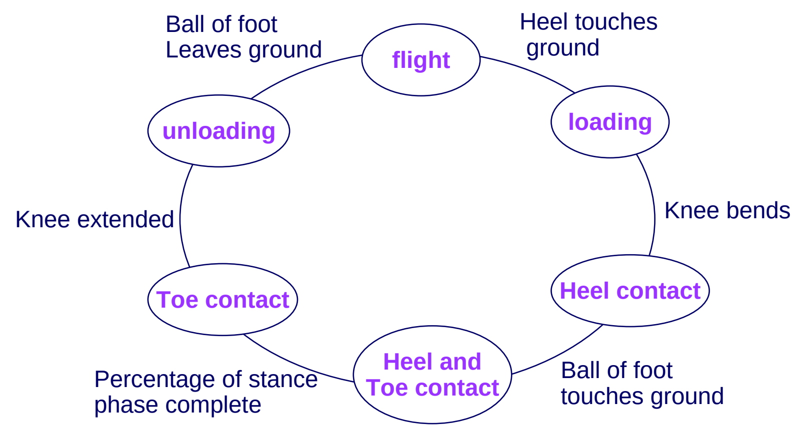
1. State machine
2. Control actions
3. Low level control

# Hierarchy of control laws

1. State machine
2. Control actions
3. Low level control



# Running state machine





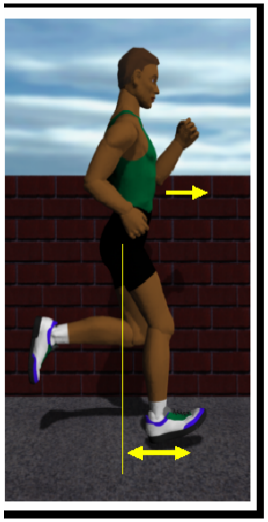

# Hierarchy of control laws

1. State machine
2. Control actions
3. Low level control



# Flight duration

# Forward Velocity

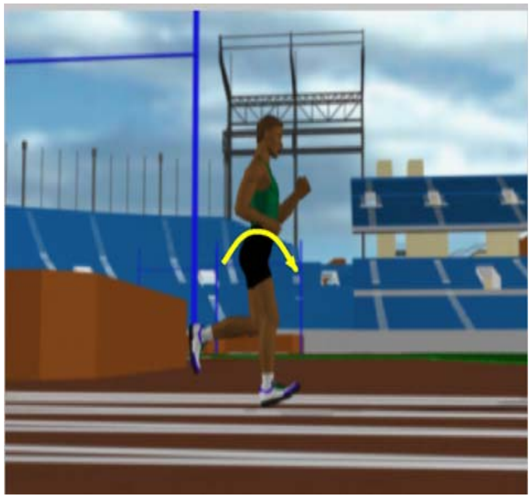




# Ground speed matching

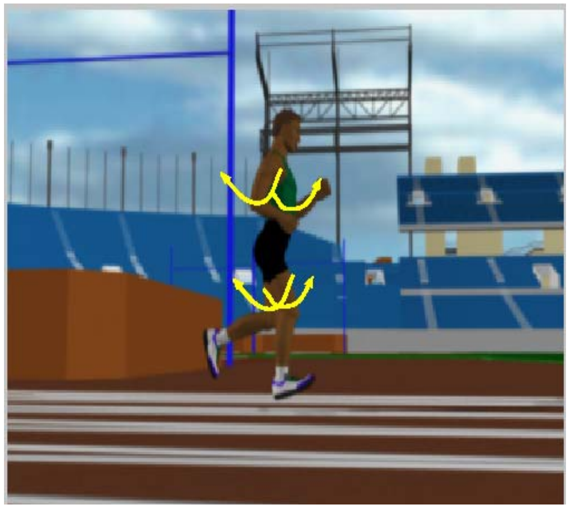


# Balance: roll, pitch, yaw





# Mirroring: hips and shoulders



# Control laws for all states

Neck: turn in desired facing direction

Shoulder: mirror hip angle

Elbow: mirror magnitude of shoulder

Wrist: constant angle

Waist: keep body upright



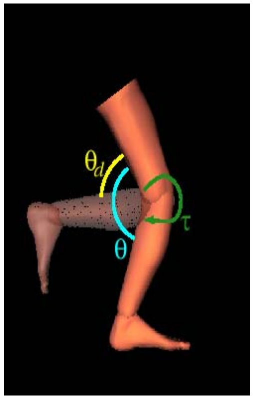
# Hierarchy of control laws

1. State machine
2. Control actions
3. Low level control



# Low level control

$$\tau = k(\theta_d - \theta) + k_v(\dot{\theta}_d - \dot{\theta})$$



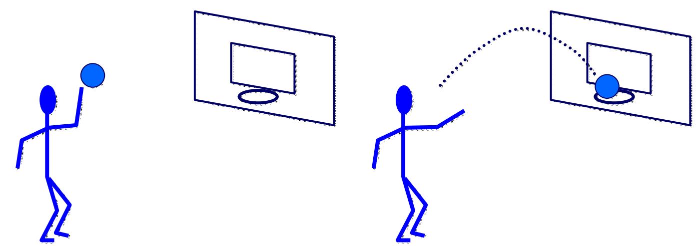



# Difference between walking and running

- Walking: double support
- Running: flight phase
- Energy transfer patterns
  - Inverted pendulum
  - Pogostick

# Spacetime constraints

- Animation is an optimal motion that achieves a given set of tasks
- Provides both realism and control





# Simulation vs. Spacetime

Forward simulation

- initial value problem

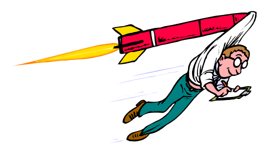
Spacetime constraints

- two-point boundary problem
- muscle forces vary as functions through time

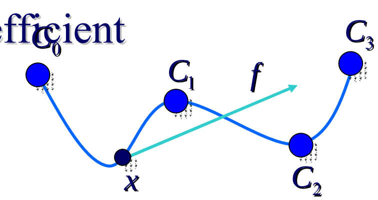


# Spacetime particle

A particle with a jet engine



- Interpolate points at specific times
- Be fuel efficient



# Equations of motion

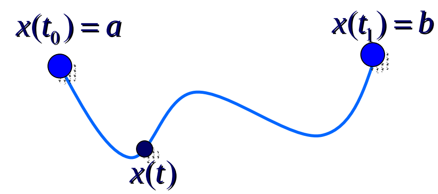
- Particle's position as a function of time *x(t)*
- Particle's mass *m*
- Time-varying jet force *f(t)*
- Constant gravitational force *mg*

$$m\ddot{x} - f - mg = 0$$



# Constraints

Fly from point *a* to point *b* in a fixed time period $t_1 - t_0$

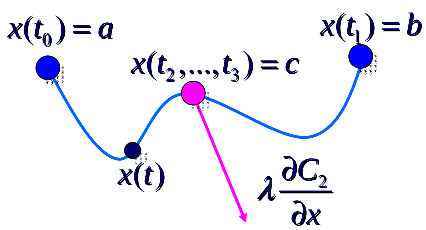





# Mechanical constraints

Constraints imposed by the environment

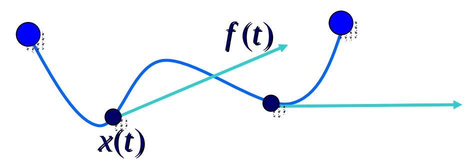
- Forces which can act to satisfy the constraint






# Jet engine "Muscle"

Force applied in arbitrary direction



# Objective function

Minimize the rate of fuel consumption

Proportional to the force magnitude integral

$$E = \int_{t_0}^{t_1} \| f(t) \|^2 \, dt$$

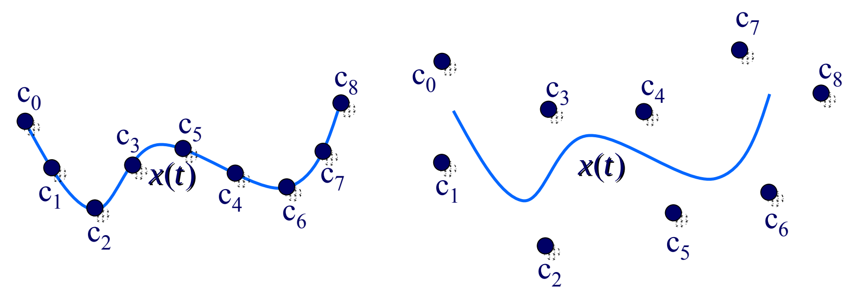


# DOF representation

$$x_i(c_0^i, \ldots, c_n^i; t)$$

$$f_j(c_0^j, \ldots, c_n^j; t)$$

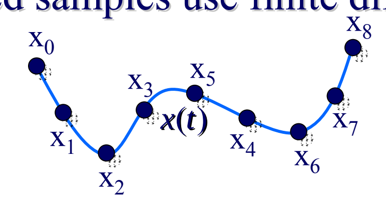
Defined in arbitrary basis:






# Computing derivatives

Discretized samples use finite differences




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

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



# Constraints formulation

- Newtonian constraint

$$n_i = m \frac{x_{i+1} - 2x_i + x_{i-1}}{h^2} - f_i - mg = 0 \quad 1 < i < n$$

- Boundary constraints

$$c_a = x_1 - a = 0$$

$$c_b = x_n - b = 0$$

- Objective function

$$E = h \sum_i \| f_i \|^2$$

$$\underset{x_i, f_i}{\text{minimize}} \quad E \qquad \text{subject to} \quad \left\{ \begin{matrix} n_i & 1 < i < n \\ c_a & \\ c_b & \end{matrix} \right\}$$

# Spacetime optimization of complex structures

When optimizing a complex mechanical structure defined by its degrees of freedom

$$[q_0, q_1, \ldots, q_n]$$

things get a lot more complicated

- Newtonian constraints become significantly more complex
- Need to convert forces into generalized forces



# Deriving Newtonian constraints

Start with Lagrange's equations of motion

$$\frac{d}{dt}\left(\frac{\partial T}{\partial \dot{q}}\right) - \frac{\partial T}{\partial q} - Q = 0$$

Derive kinetic energy *T* and generalized forces *Q*



# Muscles

Muscle force proportional to the difference between the current and desired parameter value

$$f_i = k_i\left(q_j^m - q_j\right)$$



# Importance of a good initial position

- Does not converge if the starting point is too far from the solution
- Hard to find the constraint hyper-surface
- Explosion of the number of unknowns

# Parameter and constraint explosion

- Parameter space is proportional to
  - Number of DOFs
  - Length of the optimized time period
- Constraint count is proportional to the time period
- Constraint complexity is proportional to the number of DOFs