Topic 3:

Teleoperation and dynamics

(a very brief overview)

robot control loop

- someone please sketch on the board

Topic 2: Robot Middleware

Computation

Embodiment

The Robot Control Loop

objective

G

$x_d[t]=\underline{\mathbf{AI}}(x[\hat{t}],G)$

Decision Making

$x_d[t]$

$u[t]=\underline{\mathbf{MC}}(x_d[t]-\hat{x}[t])$

Motion Control

$u[t]$

$x[t+1]=\underline{\mathbf{NEXT}}(x[t],u[t])$

Physics (Plant)

$x[t+1]$

$\hat{x}[t]$

Topic 2: Robot Middleware

$y[t]$

Perception

$\hat{x}[t]=\underline{\mathbf{PER}}(y[1:t])$

Sensing

$y[t]=\underline{\mathbf{OUT}}(x[t])$

Computation

Embodiment

Options (not mutually exclusive):
1) Teleoperation: remote control by human user(s)
2) Computer program: executing a procedural model
3) Learning: learned mappings from data/experience

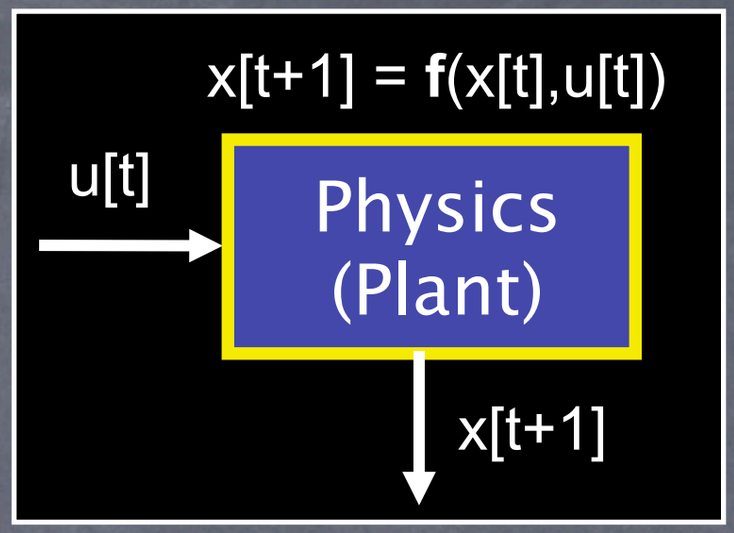
Topic 3: Kinematics and Dynamics

Simulation v. Reality

- What separates physical simulation (games, animation) from physical reality?
  - partial and noisy sensing of the real world
  - non-determinism of the real world
- Teleoperation: robots as remote control device
  - users loses situation awareness in teleop
- Understanding physics is crucial for engineering robotic systems

Physics

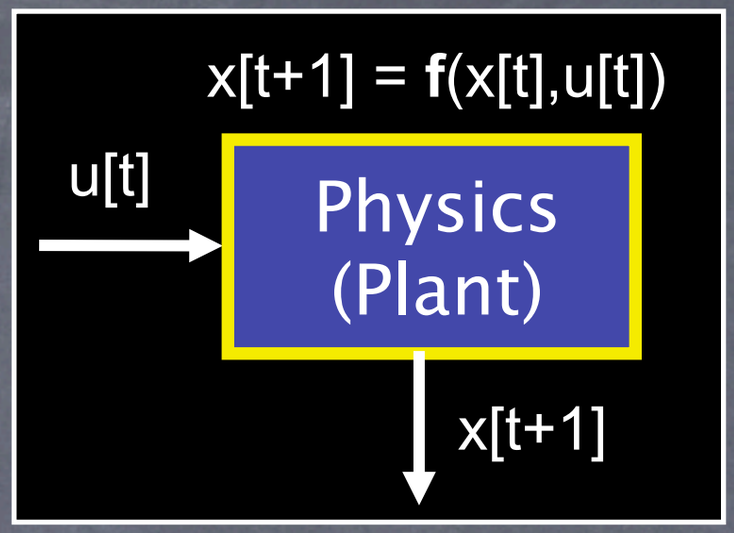
(makes the world go round)



- Predict next state (x_{t+1}) from current state (x_t)
  - $\Delta x = f(x_t, u_t)$
  - $x_{t+1} = x_t$ + delta_x
- Questions
  - How do we compute $f(x_t, u_t)$?
  - What is x_t?
  - What is u_t?

Physics

(makes the world go round)



- Predict next state (x_{t+1}) from current state (x_t)
  - $\Delta x = f(x_t, u_t)$
  - $x_{t+1} = x_t$ + delta_x
- Questions
  - How do we compute $f(x_t, u_t)$? Dynamics
  - How is x_t defined? Kinematics (motion w/o physics)
  - How do we compute u_t? Motion control

Physics: $f(x_t, u_t)$

- Do we understand the true nature of the universe?

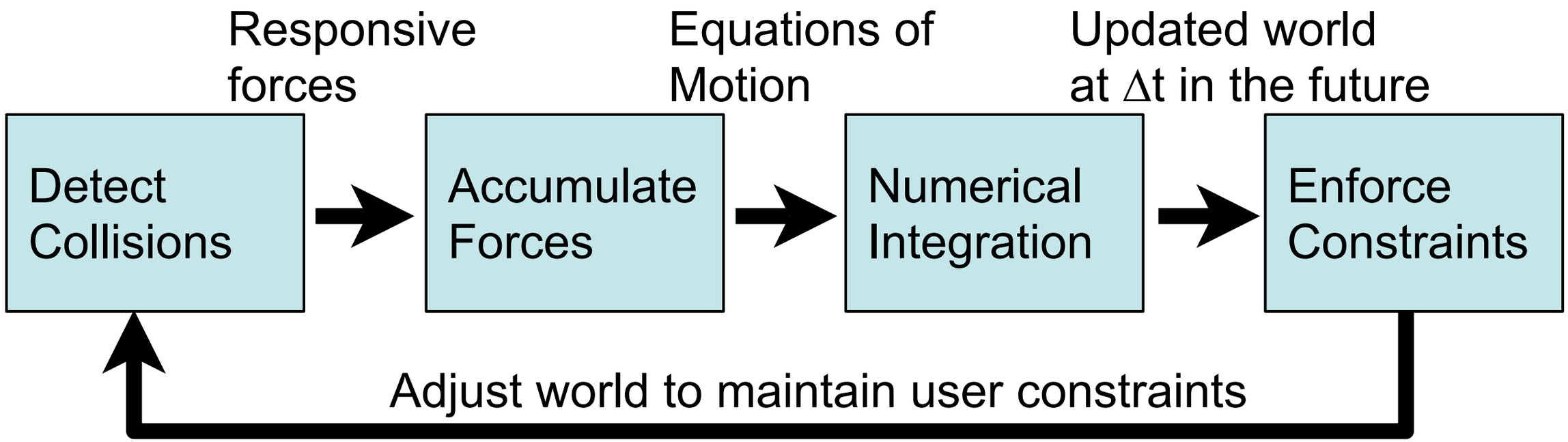
Physics: $f(x_t, u_t)$

- Do we understand the true nature of the universe? .... probably not
- But we have good approximations
  - Newtonian mechanics (3 laws of motion)
  - Quantum mechanics
  - Relativity theory
  - String theory

Physics: $f(x_t, u_t)$

- Do we understand the true nature of the universe? .... probably not
- But we have good approximations

Computational approximation (typical):



Newton's Laws of Motion

- Consider two particles, x_A and x_B, with 2D position

1. "A body persists its state of rest or of uniform motion unless acted upon by an external unbalanced force."

2. Force equals mass times acceleration

   - $F = ma = m\,(dv/dt) = m\,(d^2x/dt^2)$

3. "To every action there is an equal and opposite reaction."

   - $F_{B->A} = -F_{A->B}$

Newton's Laws of Motion

- Consider two particles, x_A and x_B, with 2D position

1. "A body persists its state of rest or of uniform motion unless acted upon by an external unbalanced force." Concept of inertia

2. Force equals mass times acceleration

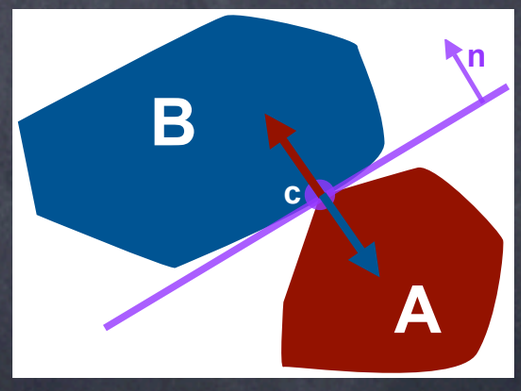
   - $F = ma = m\,(dv/dt) = m\,(d^2x/dt^2)$

Predict next state in time

3. "To every action there is an equal and opposite reaction."

   - $F_{A->B} = -F_{B->A}$

React to collisions



Newton's 2nd Law

Note: $\ddot{x} = a$

Restate f=ma as:

$$\ddot{x} = f(x, \dot{x}, t)/m$$

net force (internal,external): $F = \sum_i f_i$

Restate state as position and velocity:

$$\dot{v} = F/m$$
$$\dot{x} = v$$

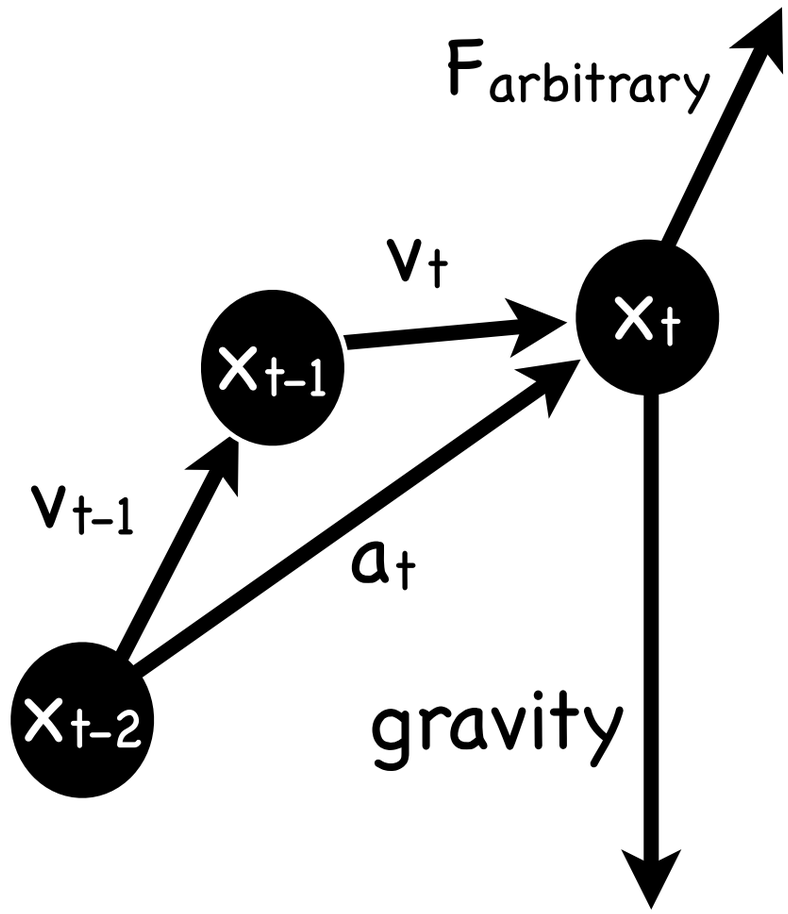
$$\frac{d}{dt}\begin{bmatrix} \dot{x} \\ \dot{v} \end{bmatrix} = \begin{bmatrix} v \\ F/m \end{bmatrix}$$

equations of motion

Discrete approximation:

$$v = \dot{x} = \frac{\mathrm{d}x}{\mathrm{d}t} \approx \frac{x_t - x_{t+\Delta t}}{\Delta t}$$

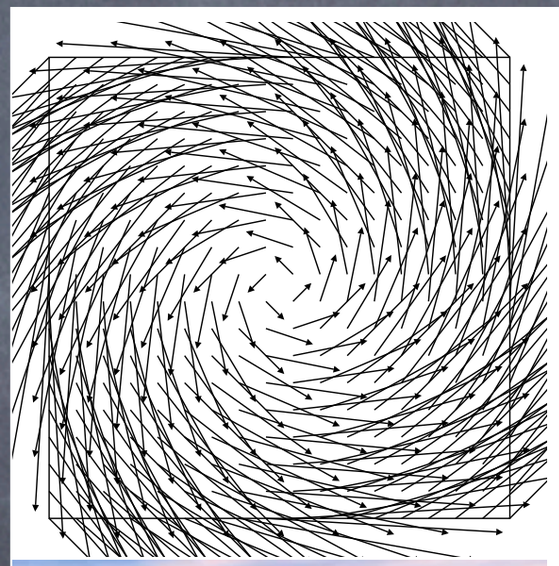
$$a = \dot{v} = \frac{\mathrm{d}v}{\mathrm{d}t} \approx \frac{v_t - v_{t+\Delta t}}{\Delta t}$$

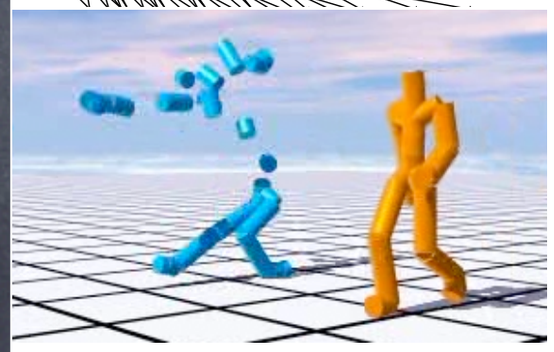


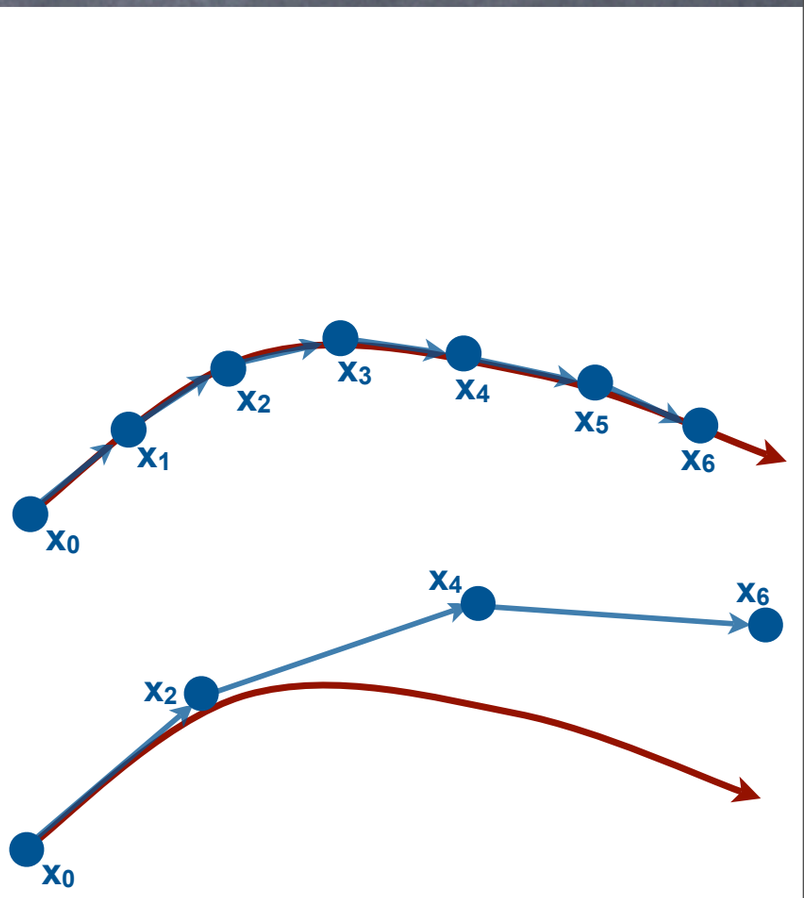
F = gravity + $F_{arbitrary}$

Solving Equations of Motion

- Initial value problem $\int_t^{t+\Delta t} f(s_t) ds_t$
- Numerical integration
  - Euler's method
    $s_{t+\Delta t} = s_t + \Delta t f(s_t)$
  - Verlet integration
- Closed form
  - Lagrangian
  - Newton-Euler



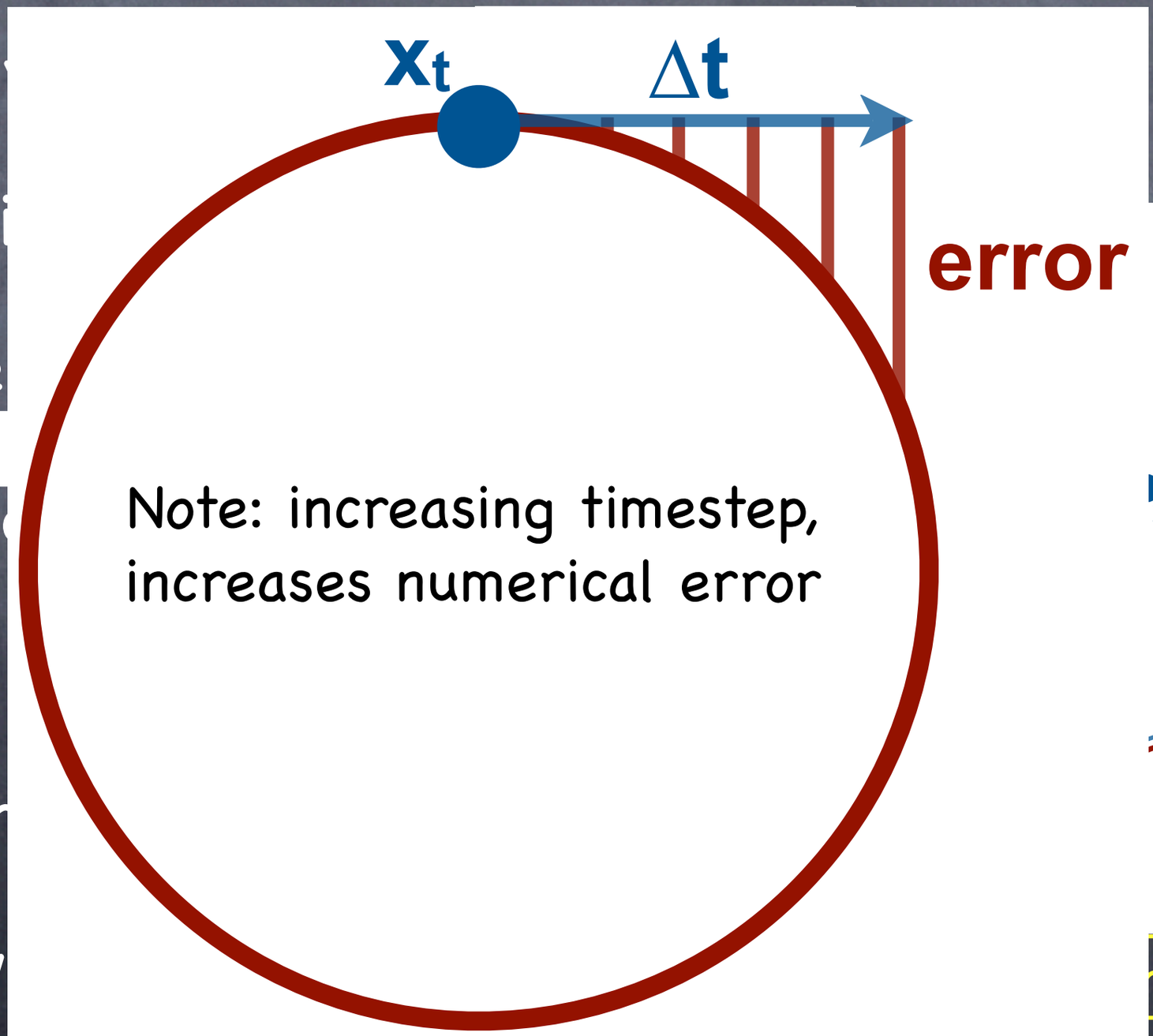




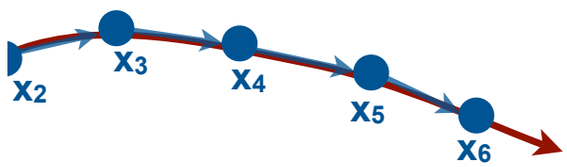
Note: generalized vs. constrained dynamics

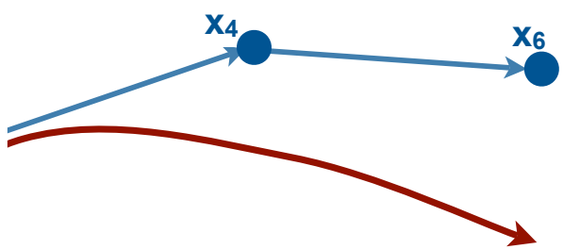
Solving Equations of Motion

- Initial
- Numeri
  - Eule
  - Verl
- Closed
  - Lagr
  - New



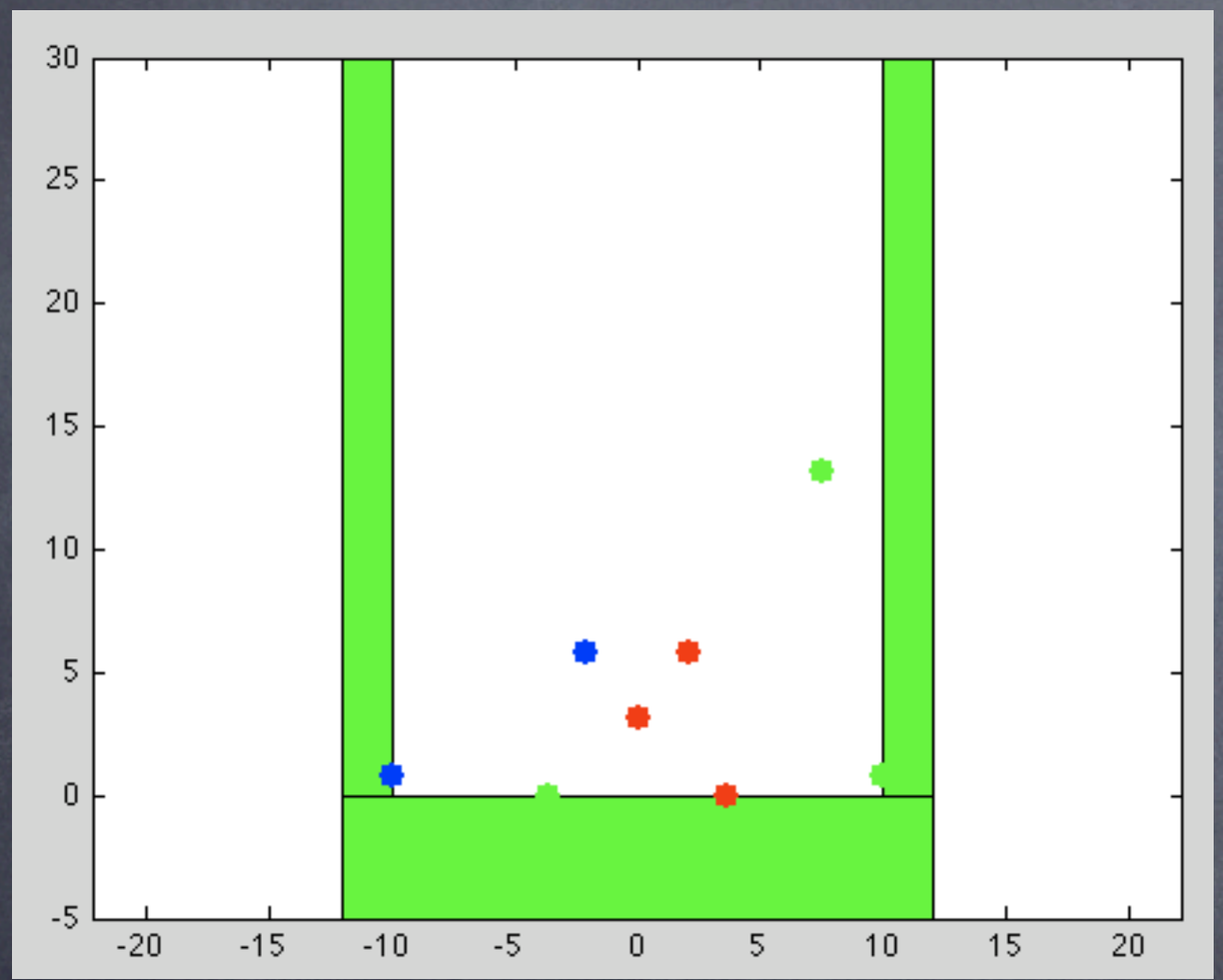
Note: increasing timestep, increases numerical error





nstrained dynamics

matlab particle sim



Far from perfect, but illustrates the basic idea

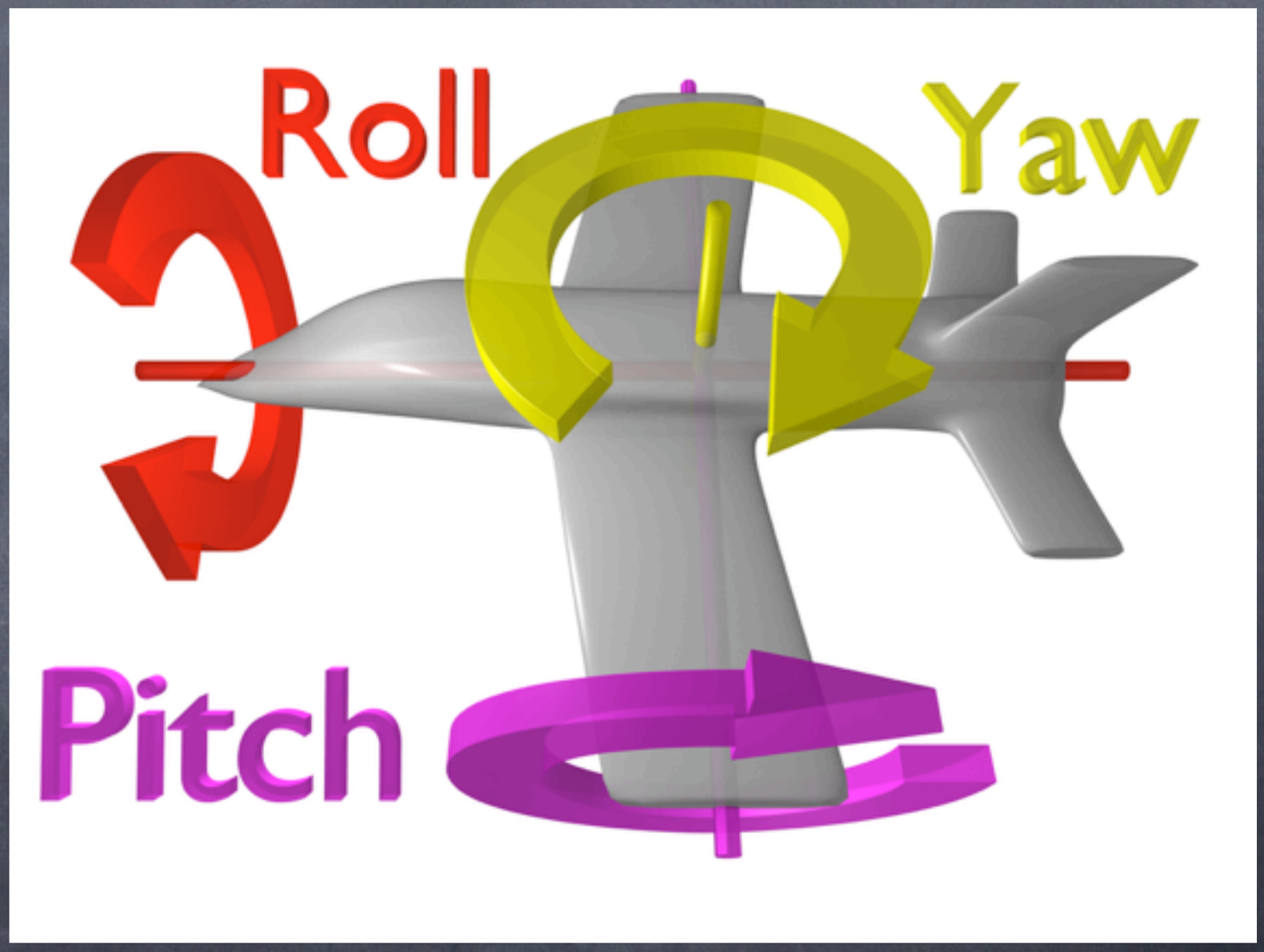
http://www.cs.brown.edu/courses/cs148/pub/particle_sim_multi.m

Physics Engines

- Physical simulation is now a commodity technology
- Simulate new plant/systems by defining state
- Available engines include
  - Open Dynamics Engine (used in Gazebo)
  - Newton
  - Havoc
  - Bullet

Defining State

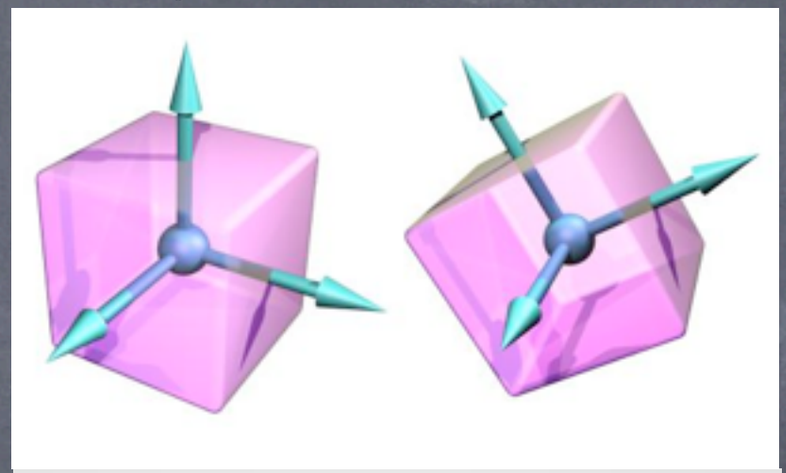
- State comprised of degrees-of-freedom (DOFs)
- DOFs describe translational and rotational axes for joints
- Particle DOFs?
- Airplane DOFs?

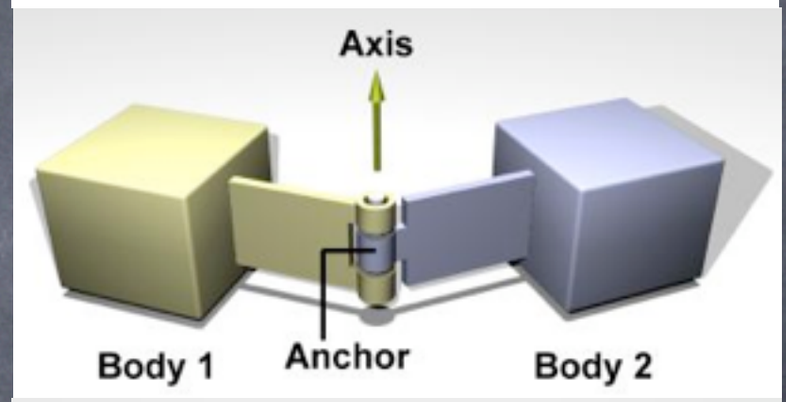


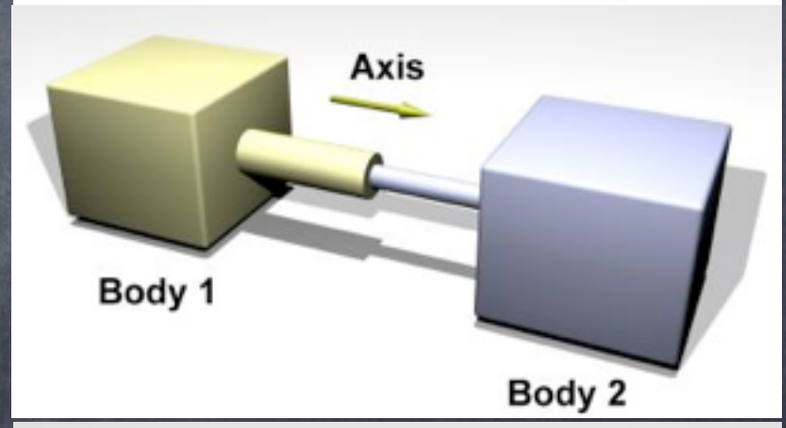
Note: particles vs. rigid bodies, COM, geom, inertia

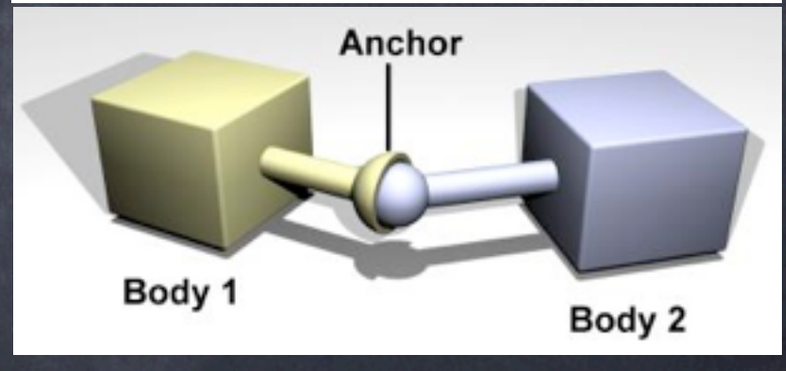
DOFs and Coordinate Spaces

- Each body has its own coordinate system
- Joints connect two rigid bodies
  - Hinge (1 rotational DOF)
  - Prismatic (1 translational DOF)
  - Ball-socket (3 DOFs)
- A motor force exerts on a DOF axis
- Linear transformations used to relate coordinate systems of bodies and joints



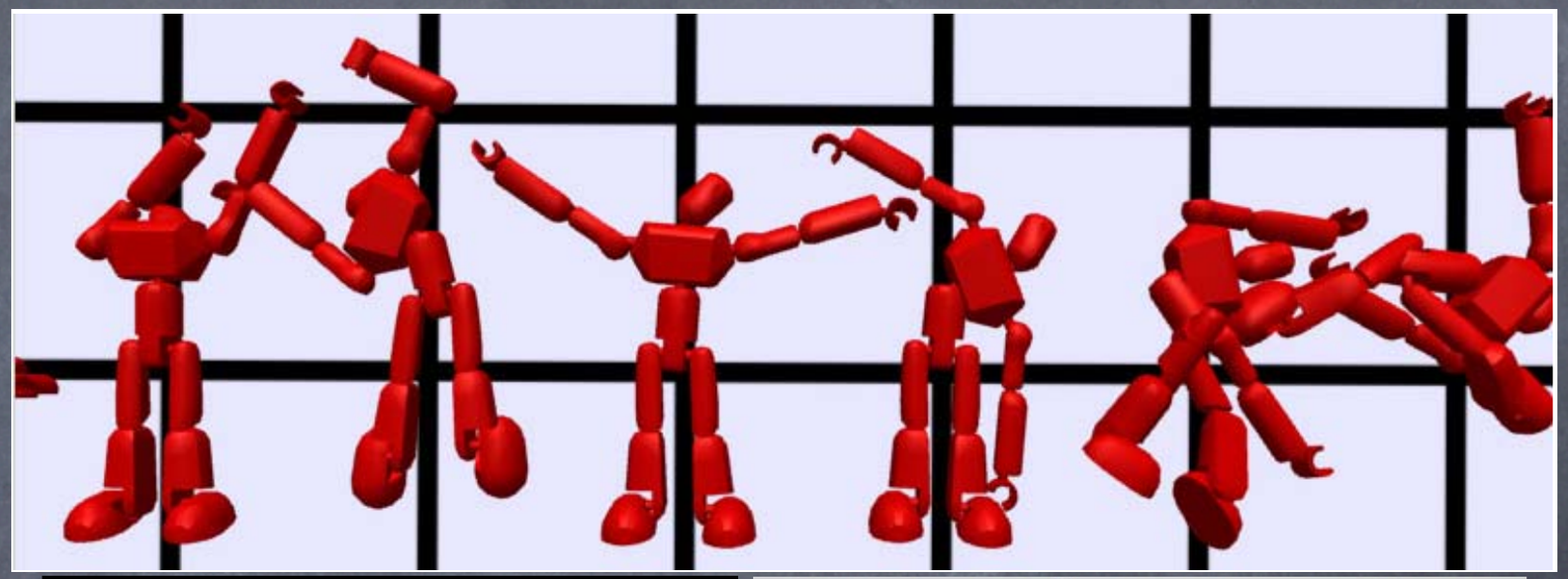


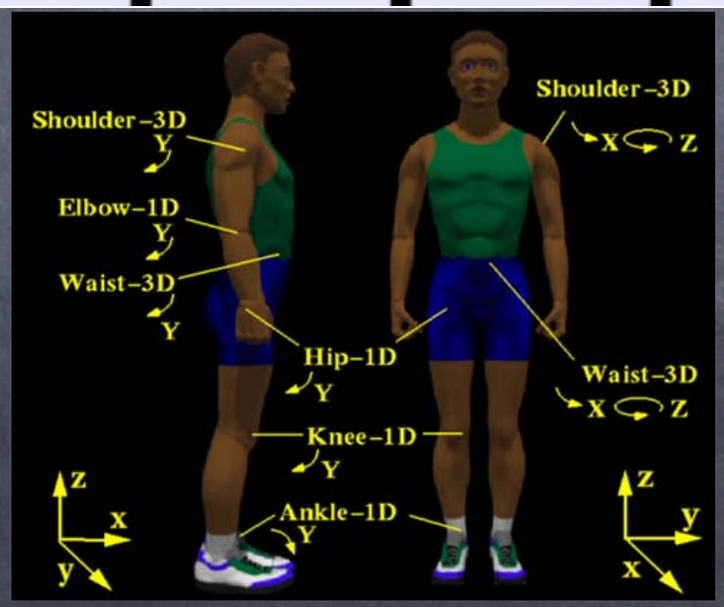


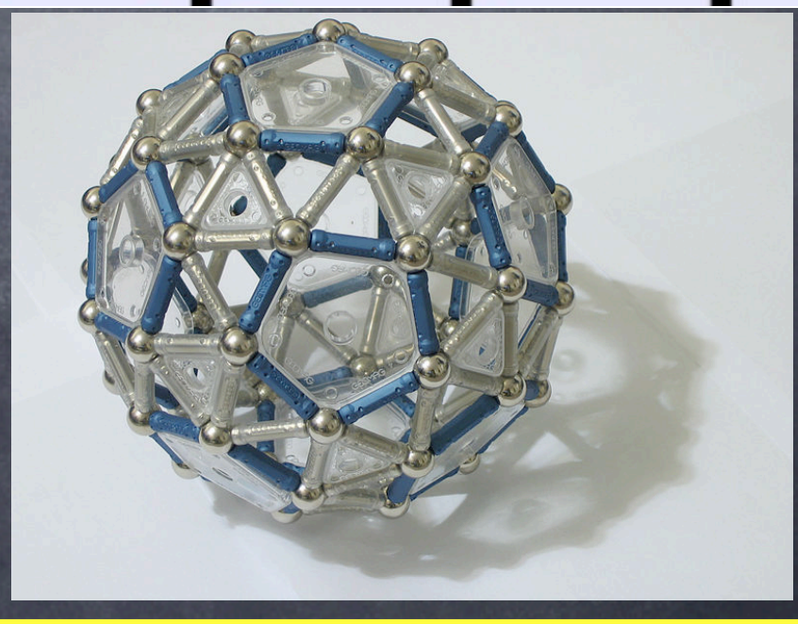


Defining State

- State comprised of degrees-of-freedom (DOFs)
- DOFs describe translational and rotational axes of system
- Humanoid DOFs?
  - joint angles



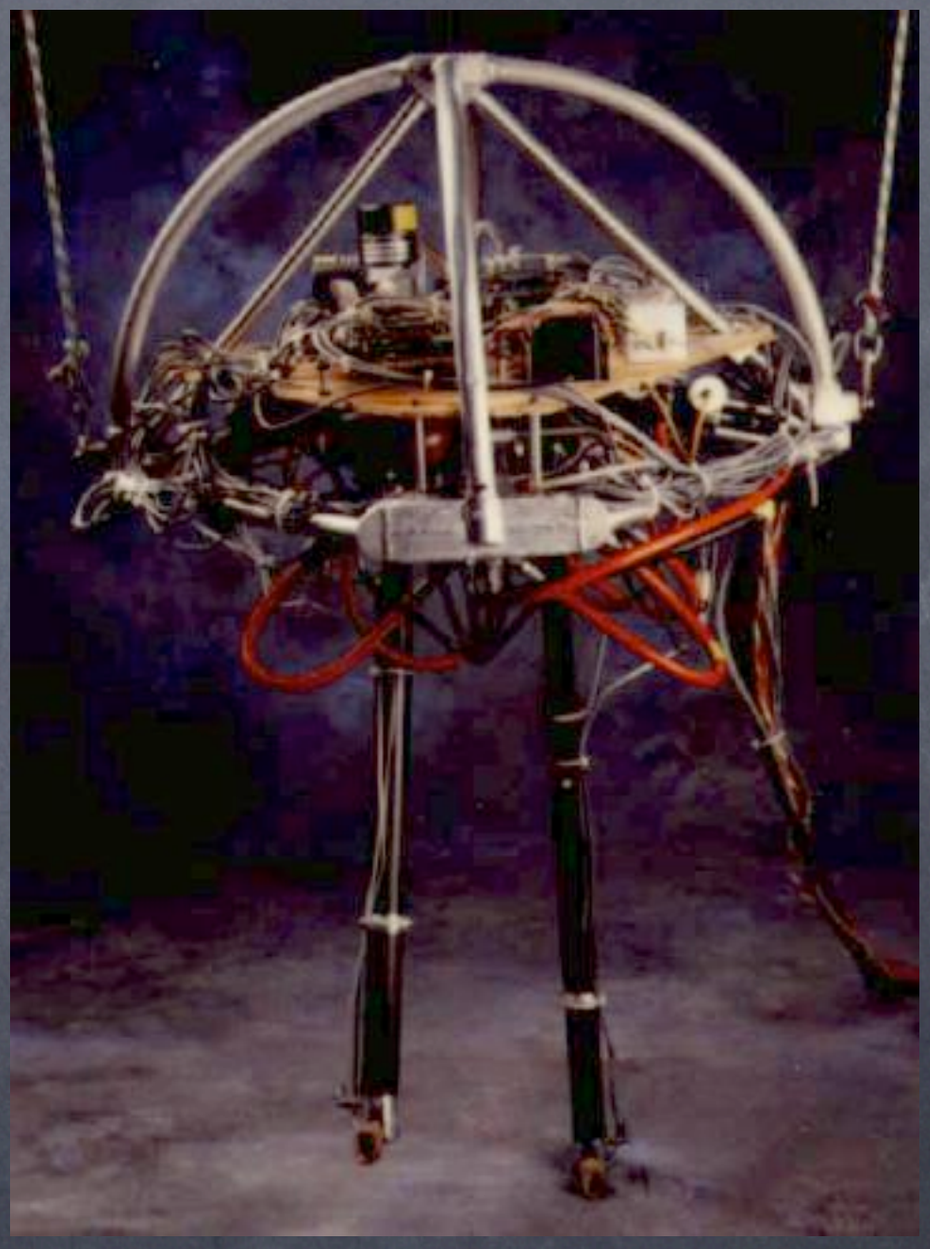




Note: types of joints, global vs. local DOFs

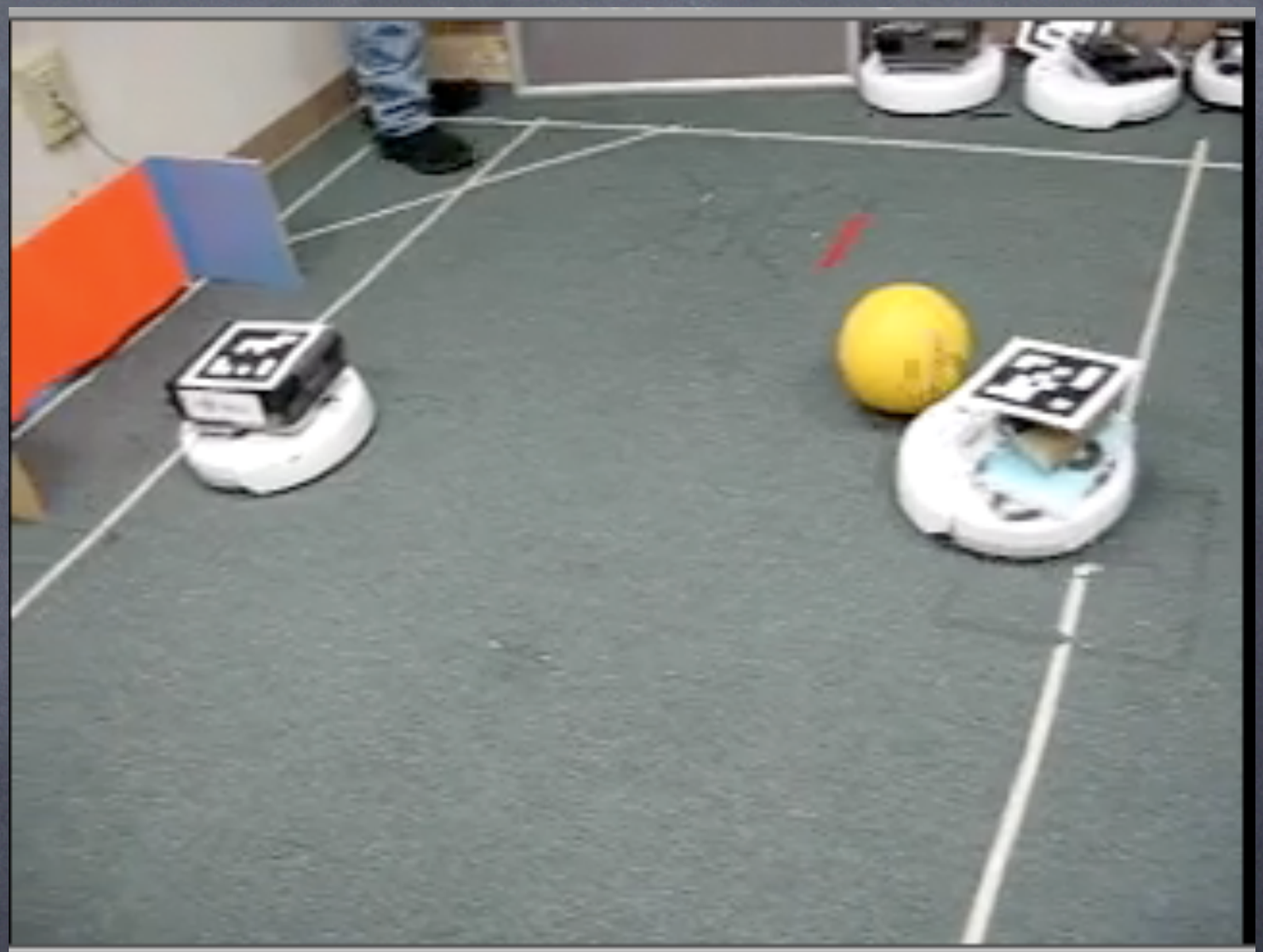
Defining State

- State comprised of degrees-of-freedom (DOFs)
- DOFs describe translational and rotational axes of system
- Biped hopper DOFs?



Defining State

- State comprised of degrees-of-freedom (DOFs)
- DOFs describe translational and rotational axes of system
- Create DOFs?



Additional issues

- Collision detection
- Articulated bodies (hierarchies of joints)
  - Denavit-hartenberg parameters
  - Matrix stack
- Constraints
  - Specifying constraints between bodies
  - Enforcing constraints during integration

Additional issues

- Mechanics of sensing
  - proprioception, exteroception
- Dynamical stability:
  - Given x_t and x_{goal}, solve for u_t
- Inverse dynamics - Lyapunov analysis
- Inverse kinematics:
  - Given hand location, solve for joint angles