CENG 732
Computer Animation

Spring 2006-2007

Week 5

Inverse Kinematics

Physically Based Rigid Body Simulation

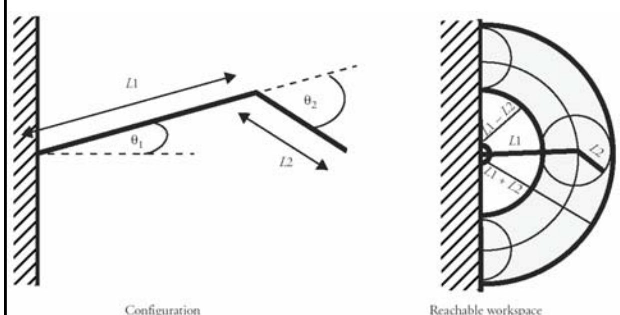
This week

- Inverse Kinematics (continued)
- Rigid Body Simulation
  - Bodies in free fall
  - Bodies in contact

Inverse Kinematics

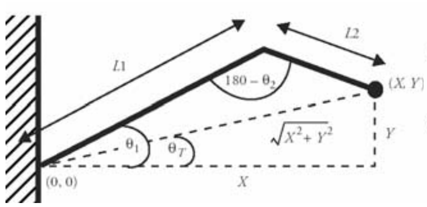
- Find the intermediate joint angles given the position and orientation of the end effector
  - Some constraints may also be given
    - E.g., joint angles in a range
- There may be no solutions
  - Overconstrained
- There may be multiple solutions
  - Underconstrained

Analytic computation for simple cases



Given (x,y) coordinate of the end point of the end effector, compute θ_1 and θ_2

Solution



$$\cos(\theta_T) = \frac{X}{\sqrt{X^2 + Y^2}}$$

$$\theta_T = \operatorname{acos}\left(\frac{X}{\sqrt{X^2 + Y^2}}\right)$$

$$\cos(\theta_1 - \theta_T) = \frac{L1^2 + X^2 + Y^2 - L2^2}{2 \cdot L1 \cdot \sqrt{X^2 + Y^2}}$$

$$\cos(180 - \theta_2) = \frac{L1^2 + L2^2 - (X^2 + Y^2)}{2 \cdot L1 \cdot L2}$$

$$\theta_1 = \operatorname{acos}\left(\frac{L1^2 + X^2 + Y^2 - L2^2}{2 \cdot L1 \cdot \sqrt{X^2 + Y^2}}\right) + \theta_T$$

$$\theta_2 = \operatorname{acos}\left(\left(\frac{L1^2 + L2^2 - (X^2 + Y^2)}{2 \cdot L1 \cdot L2}\right)\right)$$

The Jacobian

- In many complex joints however, such analytic solutions are not possible.
- Therefore we use the Jacobian matrix to find the correct joint angle increments that will lead us to the final end effector configuration
- The Jacobian matrix is a matrix of partial derivatives
  - Each entry shows how much the change in an input parameter effects an output parameter

Example Jacobian

$$x_1 = r\sin\phi\cos\theta$$
$$x_2 = r\sin\phi\sin\theta$$
$$x_3 = r\cos\phi$$

$$J_F(r,\phi,\theta) = \begin{bmatrix} \frac{\partial x_1}{\partial r} & \frac{\partial x_1}{\partial \phi} & \frac{\partial x_1}{\partial \theta} \\ \frac{\partial x_2}{\partial r} & \frac{\partial x_2}{\partial \phi} & \frac{\partial x_2}{\partial \theta} \\ \frac{\partial x_3}{\partial r} & \frac{\partial x_3}{\partial \phi} & \frac{\partial x_3}{\partial \theta} \end{bmatrix} = \begin{bmatrix} \sin\phi\cos\theta & r\cos\phi\cos\theta & -r\sin\phi\sin\theta \\ \sin\phi\sin\theta & r\cos\phi\sin\theta & r\sin\phi\cos\theta \\ \cos\phi & -r\sin\phi & 0 \end{bmatrix}.$$

Example Jacobian

$$y_1 = x_1$$
$$y_2 = 5x_3$$
$$y_3 = 4x_2^2 - 2x_3$$
$$y_4 = x_3\sin(x_1)$$

$$J_F(x_1,x_2,x_3) = \begin{bmatrix} \frac{\partial y_1}{\partial x_1} & \frac{\partial y_1}{\partial x_2} & \frac{\partial y_1}{\partial x_3} \\ \frac{\partial y_2}{\partial x_1} & \frac{\partial y_2}{\partial x_2} & \frac{\partial y_2}{\partial x_3} \\ \frac{\partial y_3}{\partial x_1} & \frac{\partial y_3}{\partial x_2} & \frac{\partial y_3}{\partial x_3} \\ \frac{\partial y_4}{\partial x_1} & \frac{\partial y_4}{\partial x_2} & \frac{\partial y_4}{\partial x_3} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 5 \\ 0 & 8x_2 & -2 \\ x_3\cos(x_1) & 0 & \sin(x_1) \end{bmatrix}.$$

Using the Jacobian

$$V = [v_x, v_y, v_z, \omega_x, \omega_y, \omega_z]^T$$

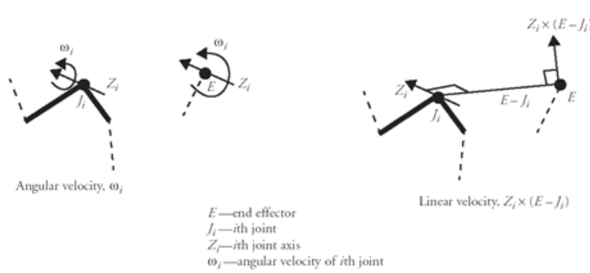
$$\dot{\theta} = [\dot{\theta}_1, \dot{\theta}_2, \dot{\theta}_3, \ldots, \dot{\theta}_n]^T$$

$$V = J(\theta)\dot{\theta}$$

$$J^{-1}V = \dot{\theta}$$

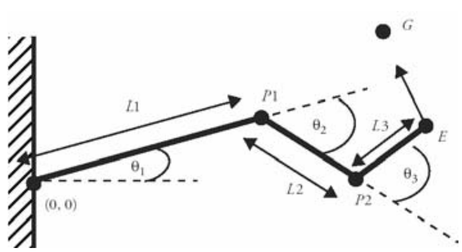
$$J = \begin{bmatrix} \frac{\partial v_x}{\partial \theta_1} & \frac{\partial v_x}{\partial \theta_2} & \cdots & \frac{\partial v_x}{\partial \theta_n} \\ \frac{\partial v_y}{\partial \theta_1} & \frac{\partial v_y}{\partial \theta_2} & \cdots & \frac{\partial v_y}{\partial \theta_n} \\ \cdots & \cdots & \cdots & \cdots \\ \frac{\partial \omega_z}{\partial \theta_1} & \frac{\partial \omega_z}{\partial \theta_2} & \cdots & \frac{\partial \omega_z}{\partial \theta_n} \end{bmatrix}$$

Computing the Jacobian

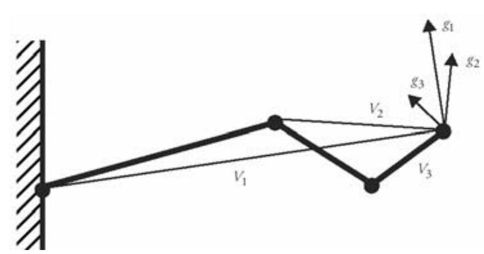


A simple example

- Assume we want to find the change in the rotation angles to get the end effector to G



Effect of changing θs



The equation

$$\begin{bmatrix}(G-E)_x\\(G-E)_y\\(G-E)_z\end{bmatrix} = \begin{bmatrix}((0,0,1)\times E)_x & (0,0,1)\times(E-P_1)_x & (0,0,1)\times(E-P_2)_x\\((0,0,1)\times E)_y & (0,0,1)\times(E-P_1)_y & (0,0,1)\times(E-P_2)_y\\((0,0,1)\times E)_z & (0,0,1)\times(E-P_1)_z & (0,0,1)\times(E-P_2)_z\end{bmatrix}\cdot\begin{bmatrix}\dot\theta_1\\\dot\theta_2\\\dot\theta_3\end{bmatrix}$$

Solving for θs

$$V = J\dot\theta$$

$$J^{-1}V = \dot\theta$$

If J is not a square matrix

- Use the pseudo-inverse to compute the joint angles

$$V = J\dot\theta$$

$$J^T V = J^T J\dot\theta$$

$$(J^T J)^{-1} J^T V = (J^T J)^{-1} J^T J\dot\theta$$

$$J^+ V = \dot\theta$$

Adding Constraints

- Define *H* as:

$$H = \sum_{i=1}^{n} \alpha_i \cdot (\theta_i - \theta_{ci})^{\psi}$$

where θ_i is the current joint angle of joint *i*
θ_{ci} is the desired joint angle for joint *i*
α_i is the desired angle gain (the higher it is the more difficult to move the joint away from the desired joint angle)

Adding Constraints

- Define *z* as the gradient of *H*:

$$z = \nabla_\theta H = \frac{dH}{d\theta} = \psi \sum_{i=1}^{n} \alpha_i \cdot (\theta_i - \theta_{ci})^{\psi - 1}$$

we are going to add control expression involving *z* to our solution. Adding the control expression to the solution will not affect the end effector's motion

Effect of the control expression

$$\dot\theta = (J^+J - I)z$$

$$V = J\dot\theta$$

$$V = J(J^+J - I)z$$

$$V = (JJ^+J - J)z$$

$$V = (J - J)z$$

$$V = 0 \cdot z$$

$$V = 0$$

Solving for θs with the control expression

$$\dot{\theta} = J^{+}V + (J^{+}J - I)\nabla_{\theta}H$$

$$\dot{\theta} = J^{+}V + (J^{+}J - I)\nabla_{\theta}H$$

$$\dot{\theta} = J^{+}V + J^{+}J\nabla_{\theta}H - I\nabla_{\theta}H$$

$$\dot{\theta} = J^{+}(V + J\nabla_{\theta}H) - \nabla_{\theta}H$$

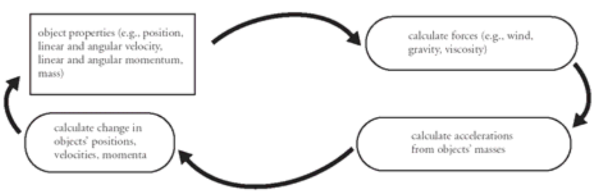
$$\dot{\theta} = J^{T}(JJ^{T})^{-1}(V + J\nabla_{\theta}H) - \nabla_{\theta}H$$

$$\dot{\theta} = J^{T}[(JJ^{T})^{-1}(V + J\nabla_{\theta}H)] - \nabla_{\theta}H$$

Rigid Body Simulation

- Reaction of rigid bodies to forces such as:
  - Gravity
  - Viscosity
  - Friction
  - Forces from collisions
  - Wind
- When applied to objects, these forces induce linear and angular accelerations

Rigid Body Simulation Cycle



The difference from standard physics is that in compute animation the motion of objects at discrete times steps is studied along with significant events and their aftermath.

Physics Primer

- Read Appendix B.6 (pages 476-488) from the text book for equation that are needed from simple physics simulation

Bodies in Free Fall

Position update when the velocity is constant

$$x(t+\Delta t) = x(t) + v(t)\cdot\Delta t$$

$$v(t+\Delta t) = v(t) + a(t)\cdot\Delta t$$

$$x(t+\Delta t) = x(t) + ((v(t) + v(t+\Delta t))/2)\cdot\Delta t$$

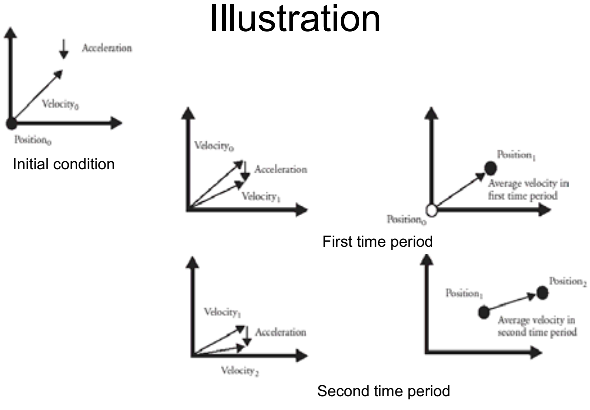
$$x(t+\Delta t) = x(t) + v(t)\cdot\Delta t + \frac{1}{2}\cdot a(t)\cdot\Delta t^{2}$$

Position update when the velocity is changing due to acceleration

A simple example

- Consider a point at (0,0) with initial velocity of (100,100) feet per second
- Gravity: (0,-32) feet per second per second
- We want to compute the position every 1/30 seconds (to generate an animation of 30 frames per second)
- We are going to compute the beginning and ending velocities for each time interval, and use the average of these velocities to update the position of the point

Illustration



The computation

$$v\frac{1}{30} = \{100, 100\} + \{0, -32\} \cdot \frac{1}{30} = \left\{100, \frac{1484}{15}\right\}$$

$$x\frac{1}{30} = \{0, 0\} + \frac{1}{2} \cdot \left(\left\{100, \frac{1484}{15}\right\} + \{100, 100\}\right) \cdot \frac{1}{30} = \left\{\frac{10}{3}, \frac{746}{225}\right\}$$

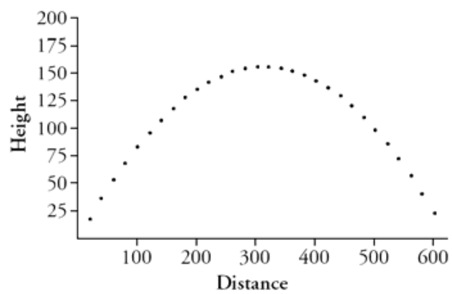
$$v\frac{1}{15} = \left\{100, \frac{1484}{15}\right\} + \{0, -32\} \cdot \frac{1}{30} = \left\{100, \frac{1468}{15}\right\}$$

$$x\frac{1}{15} = \left\{\frac{10}{3}, \frac{746}{225}\right\} + \frac{1}{2} \cdot \left(\left\{100, \frac{1468}{15}\right\} + \left\{100, \frac{1484}{15}\right\}\right) \cdot \frac{1}{30} = \left\{\frac{20}{3}, \frac{1484}{225}\right\}$$

$$v\frac{1}{10} = \left\{100, \frac{1468}{15}\right\} + \{0, -32\} \cdot \frac{1}{30} = \left\{100, \frac{484}{5}\right\}$$

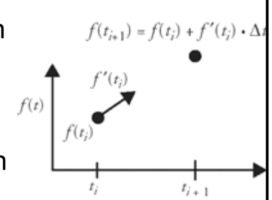
$$x\frac{1}{10} = \left\{\frac{20}{3}, \frac{1484}{225}\right\} + \frac{1}{2} \cdot \left(\left\{100, \frac{484}{5}\right\} + \left\{100, \frac{1468}{15}\right\}\right) \cdot \frac{1}{30} = \left\{10, \frac{246}{25}\right\}$$

Path of the particle

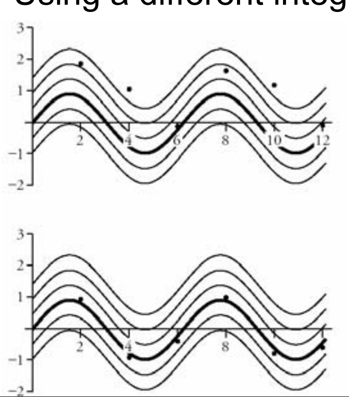


Acceleration

- In real life, the forces change as the rigid body changes its position, orientation, and velocity over the time.
- It is not the best approach to use the acceleration at the beginning of the time interval to compute the velocity at the end (known as Euler integration)



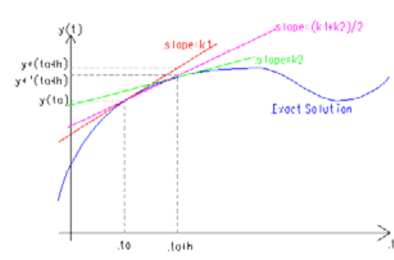
Using a different integration method



Using step sizes of 2 with the Euler method

Using step sizes of 2 with the midpoint method

RK2 Method

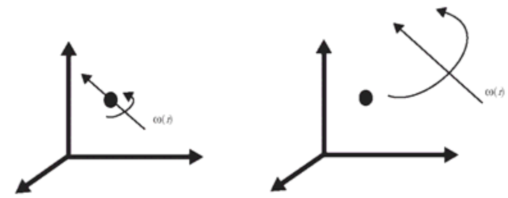


First compute the derivative of y(t) at to (which we call k1). We use k1 to get an initial estimate for y(to+h) labelled y*(to+h) . From y*(to+h) we can get an estimate for the derivative of y(t) at to+h, which we will call k2. We then use the average of these two derivatives, k3, to arrive at our final estimate of y(to+h) labelled y*'(to+h).

Rotational Motion

- For non-point objects, the mass extent of the object should be considered
- Angular velocity
  - is the rate at which the object is rotating irrespective of its linear velocity
  - the direction of the vector gives the axis of orientation
  - the magnitude gives the revolutions per unit time

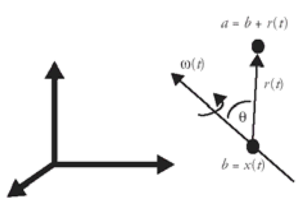
Angular Velocity



Two revolutions per second Two revolutions per second

Angular velocities are the same (but the instantaneous linear velocities are different)

Linear Velocity of a Rotating Point



$$\dot{r}(t) = \omega(t) \times r(t)$$

$$|\dot{r}(t)| = |\omega(t)||r(t)|\sin\theta$$

Center of Mass

- If mass values are provided on some discrete points on the object (i.e. vertices), the total mass and the center of mass is given by

$$M = \Sigma m_i$$

$$x(t) = \frac{\Sigma m_i q_i(t)}{M}$$

Rotating Objects

- The linear velocity of a point on a rotating object

$$\dot{q}(t) = \omega(t) \times (q(t) - x(t)) + v(t)$$

Forces

- Linear force

$$F = m \cdot a \qquad F(t) = \Sigma f_i(t)$$

$$a = F/m$$

- Torque

$$\tau_i(t) = (q(t) - x(t)) \times f_i(t)$$

$$\tau(t) = \Sigma \tau_i(t)$$

Linear Momentum

$$p = m \cdot v$$

$$P(t) = \Sigma m_i \dot{q}_i(t)$$

$$P(t) = M \cdot v(t)$$

$$\dot{P}(t) = M \cdot \dot{v}(t) = F(t)$$

Angular Momentum

$$L(t) = \Sigma((q(t) - x(t)) \times m_i \cdot (\dot{q}(t) - v(t)))$$
$$= \Sigma(R(t)q \times m_i \cdot (\omega(t) \times (q(t) - x(t))))$$
$$= \Sigma(m_i \cdot (R(t)q \times (\omega(t) \times R(t)q)))$$

$$\dot{L}(t) = \tau(t)$$

Inertia Tensor

- The matrix that describes the distribution of an object's mass in space.

$$I_{\text{object}} = \begin{bmatrix} I_{xx} & I_{xy} & I_{xz} \\ I_{xy} & I_{yy} & I_{yz} \\ I_{xz} & I_{yz} & I_{zz} \end{bmatrix}$$

$$I_{xx} = \Sigma m_i \cdot (y_i^2 + z_i^2) \qquad I_{xy} = \Sigma m_i \cdot x_i \cdot y_i$$

$$I_{yy} = \Sigma m_i \cdot (x_i^2 + z_i^2) \qquad I_{xz} = \Sigma m_i \cdot x_i \cdot z_i$$

$$I_{zz} = \Sigma m_i \cdot (x_i^2 + y_i^2) \qquad I_{yz} = \Sigma m_i \cdot y_i \cdot z_i$$

Inertia Tensor of a Rotated Object

$$I(t) = R(t) I_{\text{object}} R(t)^T$$

Summing up all together

- An object's state vector

$$S(t) = \begin{bmatrix} x(t) \\ R(t) \\ P(t) \\ L(t) \end{bmatrix} \qquad \frac{d}{dt}S(t) = \frac{d}{dt}\begin{bmatrix} x(t) \\ R(t) \\ P(t) \\ L(t) \end{bmatrix} = \begin{bmatrix} v(t) \\ \omega(t)^* R(t) \\ F(t) \\ \tau(t) \end{bmatrix}$$

Bodies in Contact

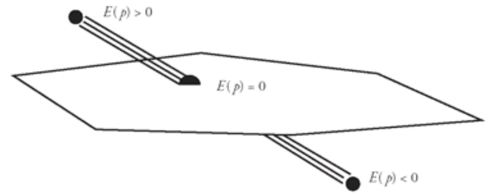
- Collision
  - Both kinematic and dynamic components
    - Kinematic: Do determine whether two objects collide or not. Only dependent on the position and orientations of the objects and how they change over time
    - Dynamic: What happens after collision, what forces are exchanged, how do they affect objects' motions

Collision handling

- Kinematic response
- Take actions after the collision occurs (the penalty method)
- Back up time to the first instant the collision occurs and determine the appropriate response

Kinematic Response

- Particle-Plane collision



$$E(p) = a \cdot x + b \cdot y + c \cdot z + d$$

Kinematic Response

- Particle's position is computed at every time step (particle is moving at a constant speed)

$$p(t_i) = p(t_{i-1}) + \partial t \cdot v_{\text{ave}}(t)$$

when $E(p(t_i)) \leq 0$ we understand that the particle has collided with the plane in the time interval t_{i-1} and t_i.

Kinematic Response

- When collision is detected, the component of the velocity vector in the normal direction is negated by subtracting it twice from the original velocity vector.
- To model the loss of energy during collision a damping factor 0<k<1 is multiplied with the normal component when it is subtracted the second time

$$v(t_{i+1}) = v(t_i) - v(t_i) \cdot N - k \cdot v(t_i) \cdot N$$
$$= v(t_i) - (1 + k) \cdot v(t_i) \cdot N$$

Kinematic Response