PENALTY-BASED MULTIBODY ANIMATION

Multibody animation

- Penalty-based
- Impulse-based
- Constrained-based

Multibody animation

- Penalty-based (Chapter 5)
- Impulse-based (Chapter 6)
- Constrained-based (Chapter 7)

Basic idea

- Rigid bodies do not penetrate each other.
- A spring-damper system is used to penalize penetration.

Motion of a single rigid body

- r: center of mass
- q: orientation
- v: linear velocity
- w: angular velocity

$$\frac{d}{dt}\begin{bmatrix} r \\ q \\ v \\ \omega \end{bmatrix} = \begin{bmatrix} v \\ \frac{1}{2}\omega q \\ a \\ \alpha \end{bmatrix},$$

$$a = \frac{\boldsymbol{F}}{m} \quad \text{and} \quad \alpha = \boldsymbol{I}^{-1}(\boldsymbol{\tau} + \omega \times \boldsymbol{I}\omega)$$

linear acceleration   angular acceleration

Motion of a single rigid body

GOAL

- Find the force and torque
- Integrate to get to position

- r: center of mass
- q: orientation
- v: linear velocity
- w: angular velocity

$$\frac{d}{dt}\begin{bmatrix} r \\ q \\ v \\ \omega \end{bmatrix} = \begin{bmatrix} v \\ \frac{1}{2}\omega q \\ a \\ \alpha \end{bmatrix},$$

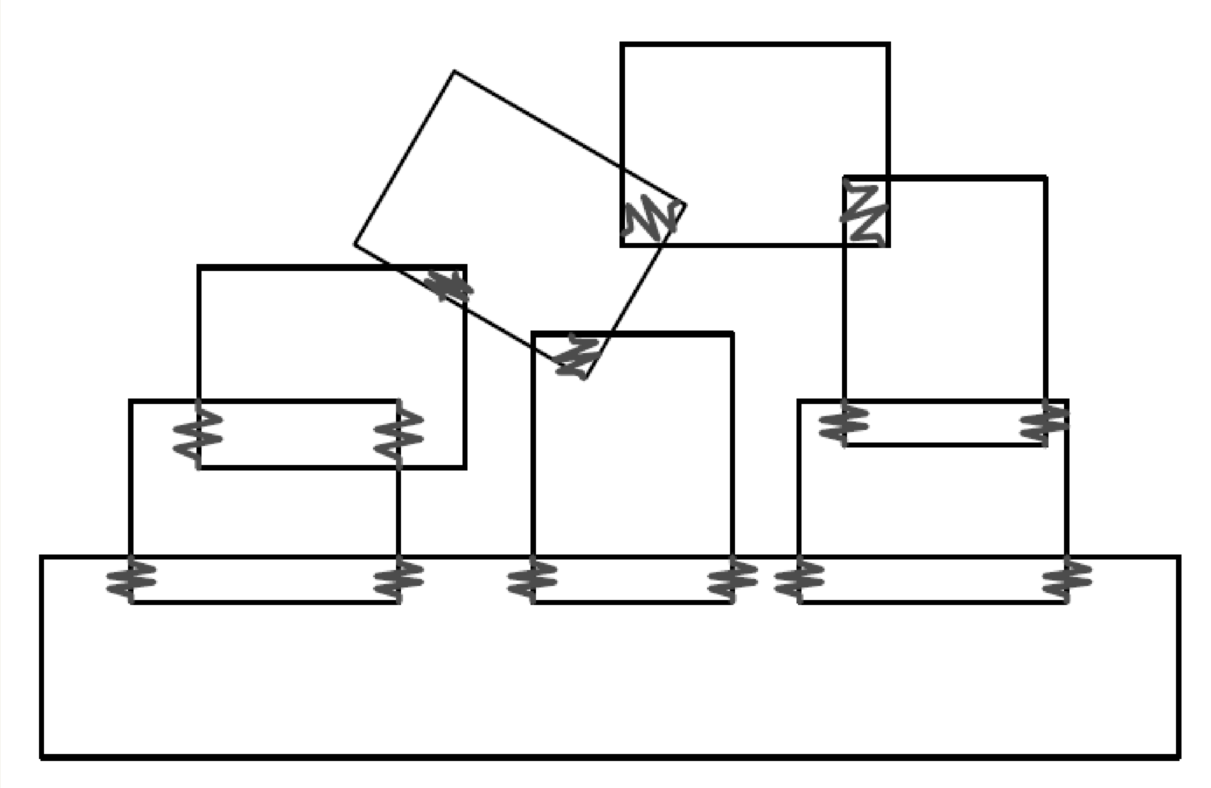
$$a = \frac{F}{m} \quad \text{and} \quad \alpha = I^{-1}\left(\tau + \omega \times I\omega\right)$$

linear acceleration

angular acceleration

Springs for the penalty

- Forces might come from contact with other rigid bodies.
- To penalize penetration, we can insert springs with a rest-length of zero at every contact point.
- The larger the penetration, the bigger the spring.



Simulation loop

- Detect contact points (run collision detection)
- Compute and accumulate spring forces
- Integrate equations of motion forward in time

How to compute spring forces

$$\boldsymbol{F}_j = \underbrace{(-kd_k}_{\text{spring term}} - \underbrace{bu_k \cdot n_k)}_{\text{damper term}}\, n_k,$$

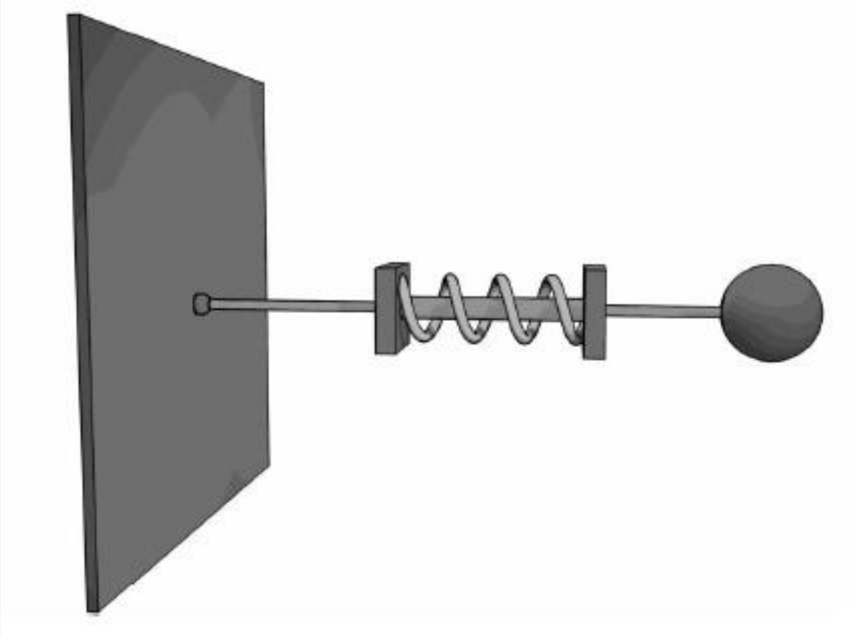
- k-th contact point
- n_k: normal between the two objects
- d_k: penetration depth
- u_k: relative contact velocity

Harmonic oscillator

- Undamped oscillator:

$$m\ddot{x} + kx = 0$$

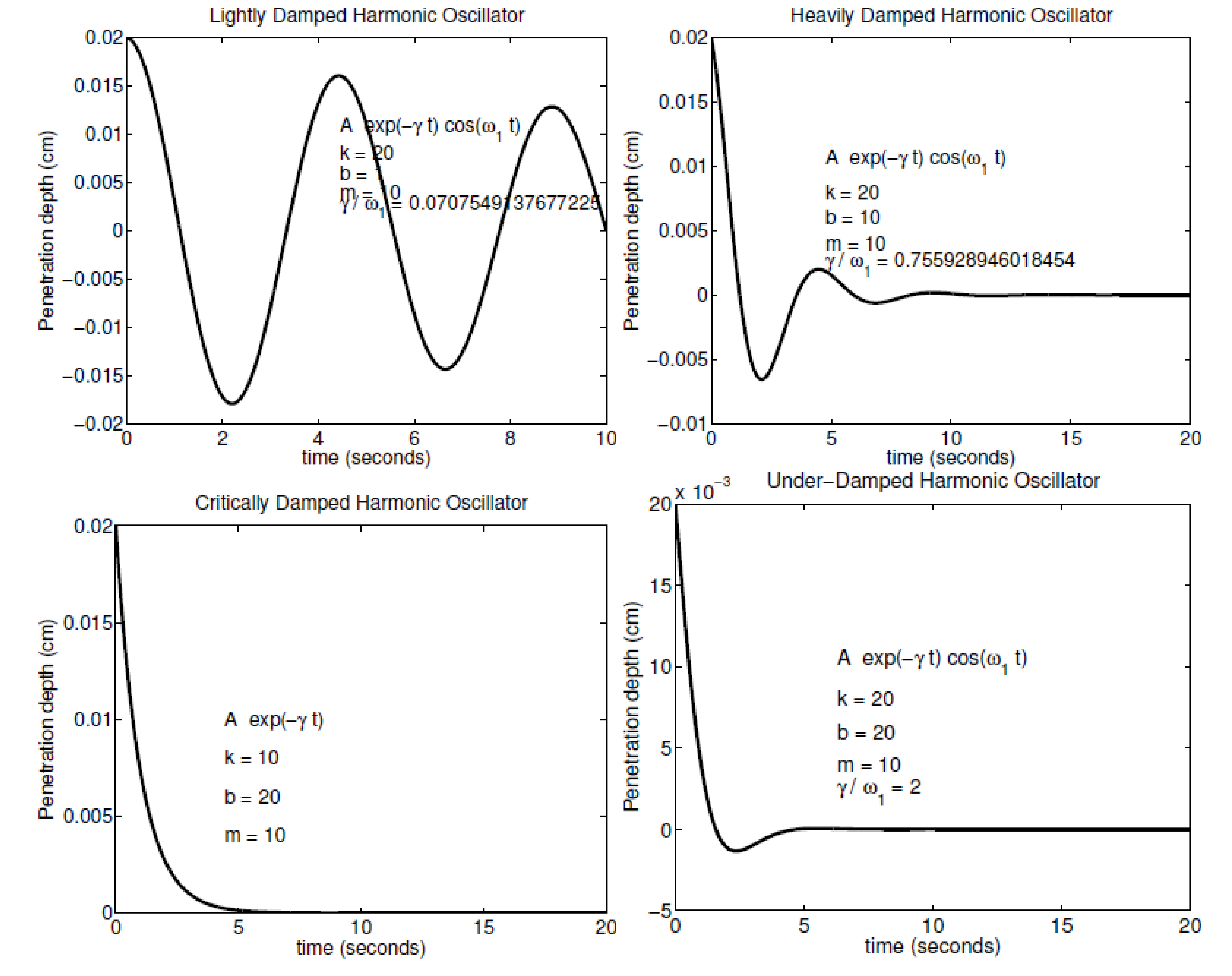
$$x = A\cos(\omega_0 t + \phi)$$ s.t. $$\omega_0 = \sqrt{\frac{k}{m}}$$



- Damped oscillator:

$$m\ddot{x} + bx' + kx = 0 \rightarrow \ddot{x} + \gamma\dot{x} + \omega_0^2 x = 0.$$

$$x = A\exp\left(-\frac{\gamma}{2}t\right)\cos(\omega_1 t + \phi)$$ s.t. $$\omega_1 = \sqrt{\omega_0^2 - \frac{\gamma^2}{4}}$$



Lightly Damped Harmonic Oscillator
A exp(−γ t) cos(ω1 t)
k = 20
b = 1
m = 10
γ / ω1 = 0.0707549137677225
Penetration depth (cm)
time (seconds)
Heavily Damped Harmonic Oscillator
A exp(−γ t) cos(ω1 t)
k = 20
b = 10
m = 10
γ / ω1 = 0.755928946018454
Penetration depth (cm)
time (seconds)
Critically Damped Harmonic Oscillator
A exp(−γ t)
k = 10
b = 20
m = 10
Penetration depth (cm)
time (seconds)
Under−Damped Harmonic Oscillator
x 10−3
A exp(−γ t) cos(ω1 t)
k = 20
b = 20
m = 10
γ / ω1 = 2
Penetration depth (cm)
time (seconds)

How to select *k* and *b*?

- Requirements:
  - No oscillation
  - Decay to zero within one frame

decay fraction of amplitude

time interval between two frames

mass

Algorithm pick-parameter-values($\Delta t, \varepsilon, m$)
  $n = \lceil -\ln \varepsilon \rceil$
  $\tau = \Delta t / n$
  $b = 2m/\tau$
  $k = m/\tau^2$
End algorithm

Solving harmonic oscillator numerically

$$x(t) = A\exp\left(-\frac{\gamma}{2}t\right) \longrightarrow \dot{x}(t) = -\frac{\gamma}{2}x(t)$$

- Explicit Euler: $$x_{i+1} = \left(1 + h\frac{-\gamma}{2}\right)x_i$$

- Implicit Euler: $$x_{i+1} = \frac{x_i}{1 + h\frac{\gamma}{2}}$$