Today

- Rigid-body dynamics
- Articulated systems
A Rigid Body

- A solid object that does not deform
  - Consists of infinite number of infinitesimal mass points...
  - ...that share a single RB transformation
    - Rotation + Translation (no shear or scale)
      \[ x^W = R \cdot x^L + t \]
    - Rotation and translation vary over time
  - Limit of deformable object as \( k_s \to \infty \)

A Rigid Body

In 2D:
- Translation 2 “directions”
- Rotation 1 “direction”
- 3 DOF Total

In 3D:
- Translation 3 “directions”
- Rotation 3 “direction”
- 6 DOF Total

Translation and rotation are decoupled

2D is boring... we'll stick to 3D from now on...
Translational Motion

Just like a point mass:
\[ \dot{p} = v \]
\[ \dot{v} = a = f/m \]

Note: Recall discussion on integration...

Rotational Motion

Rotation gets a bit odd, as well see...

Rotational “position” \( R \)
- Rotation matrix
- Exponential map
- Quaternions

Rotational velocity \( \omega \)
- Stored as a vector (Also called angular velocity...)
- Measured in radians / second
Rotational Motion

Kinetic energy due to rotation:

“Sum energy (from rotation) over all points in the object”

\[ E = \int_{\Omega} \frac{1}{2} \rho \dot{x} \cdot \dot{x} \, du \]

\[ E = \int_{\Omega} \frac{1}{2} \rho (\omega \times x) \cdot (\omega \times x) \, du \]

Rotational Motion

momentum to linear momentum derived from rotational energy

\[ H = \int_{\Omega} \rho x \times \dot{x} \, du \]

\[ H = \int_{\Omega} \rho x \times (\omega \times x) \, du \]

\[ H = \left( \int_{\Omega} \cdots \, du \right) \omega \]

“Inertia Tensor” not identity matrix...
Inertia Tensor

\[ I = \int_{\Omega} \rho \begin{bmatrix} y^2 + z^2 & -xy & -xz \\ -xy & z^2 + x^2 & -yz \\ -xz & -yz & x^2 + y^2 \end{bmatrix} \, du \]

See example for simple shapes at http://scienceworld.wolfram.com.physics/MomentofInertia.html

Can also be computed from polygon models by transforming volume integral to a surface one. See paper/code by Brian Mirtich.

Rotational Motion

Conservation or momentum:

\[ H^W = I^W \omega^W \]
\[ H^W = R I^L R^T \omega^W \]
\[ H^W = \dot{R} I^L R^T \omega^W + R I^L R^T \omega^W + R I^L R^T \alpha^W \]
\[ H^W = 0 \]
\[ \dot{R} = \omega \times R \]
\[ \alpha^W = (R I^L R^T)^{-1} (-\omega^W \times H^W) \]

In other words, things wobble when they rotate.
Rotational Motion

\[ \dot{R} = [\omega \times] R \]
\[ \dot{\omega} = \alpha \]

Figure is a lie if this really is a sphere...

Take care when integrating rotations, they need to stay rotations.

\[ \alpha^W = (RI^LRT)^{-1} \left( (-\omega^W \times H^W) + \tau \right) \]
\[ \tau = f \times x \]

Couples

- A force / torque pair is a couple
  - Also a wrench (I think)
- Many couples are equivalent
Constraints

- Simples method is to use spring attachments
  - Basically a penalty method
  - Spring strength required to get good results may be unreasonably high
    - There are ways to cheat in some contexts...

Constraints

- Articulation constraints
  - Spring trick is an example of a full coordinate method
    - Better constraint methods exist
  - Reduced coordinate methods use DOFs in kinematic skeleton for simulation
    - Much more complex to explain

- Collisions
  - Penalty methods can also be used for collisions
  - Again, better constraint methods exist
Suggested Reading


