Data-driven computer animation

Tutorial 6: Rigid Body Simulation

Prof. Taku Komura
TAs: Zhouyingcheng Liao(zycliao@cs.hku.hk), Mingyi Shi(myshi@cs.hku.hk)

Assignment 3

- Implement a rigid body simulator using Python Taichi Library

Rigid body simulation

Particles Simulation

$$\frac{d}{dt}\mathbf{Y}(t) = \frac{d}{dt}\begin{pmatrix} x(t) \\ v(t) \end{pmatrix} = \begin{pmatrix} v(t) \\ F(t)/m \end{pmatrix}$$

Particles Simulation

$$\frac{d}{dt}\mathbf{Y}(t) = \frac{d}{dt}\begin{pmatrix} x(t) \\ v(t) \end{pmatrix} = \begin{pmatrix} v(t) \\ F(t)/m \end{pmatrix}$$

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

Rigid Body Concepts

Body Space

- Our goal is to develop an analogue to equation to particles simulation for rigid body.

$$\mathbf{Y}(t) = \begin{array}{l} \left(\begin{array}{c} x(t) \\ R(t) \\ P(t) \\ L(t) \end{array} \right) \end{array} \begin{array}{l} \text{Position} \\ \text{Orientation} \\ \text{Linear Momentum} \\ \text{Angular Momentum} \end{array}$$

$$\frac{d}{dt}\mathbf{Y}(t) = \frac{d}{dt}\left(\begin{array}{c} x(t) \\ R(t) \\ P(t) \\ L(t) \end{array} \right)$$

Rigid Body Concepts

- We assume that a rigid body comprises of N particles.

Rigid Body Concepts

Mass and Centre of Mass

$$\frac{d}{dt}\mathbf{Y}(t) = \frac{d}{dt}\begin{pmatrix} x(t) \\ R(t) \\ P(t) \\ L(t) \end{pmatrix}$$

- $M = \sum m_i$, rigid body mass
- $x(t) = \dfrac{\sum m_i r_i}{M}$, centre of mass

Rigid Body Concepts

Linear Velocity

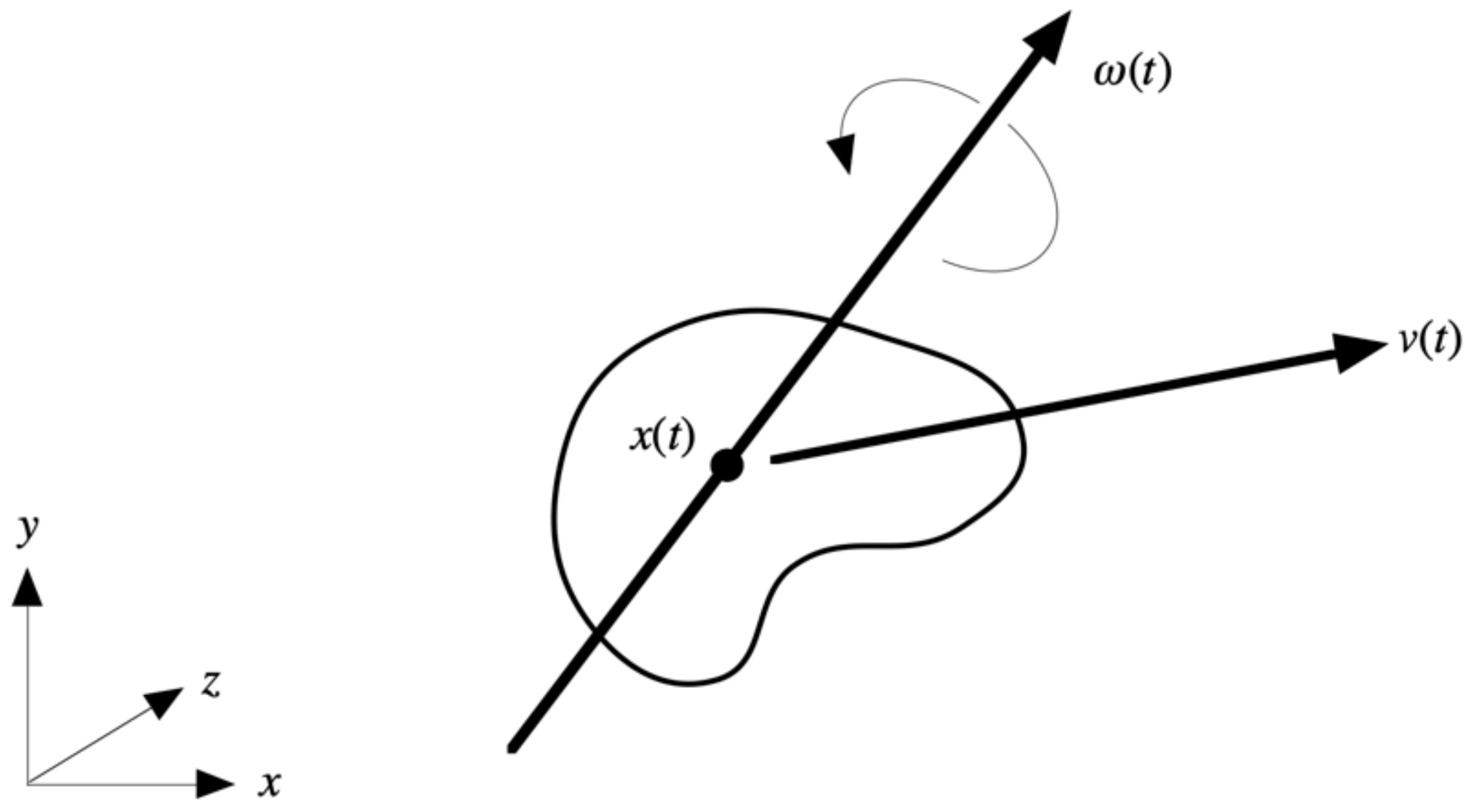
$$\frac{d}{dt}\mathbf{Y}(t) = \frac{d}{dt}\begin{pmatrix} x(t) \\ R(t) \\ P(t) \\ L(t) \end{pmatrix} = \begin{pmatrix} \mathbf{v}(t) \\ \\ \\ \\ \end{pmatrix}$$

- $\dfrac{dx(t)}{dt} = v(t)$

Rigid Body Concepts
Angular Velocity

$$\frac{d}{dt}\mathbf{Y}(t) = \frac{d}{dt}\begin{pmatrix} x(t) \\ R(t) \\ P(t) \\ L(t) \end{pmatrix} = \begin{pmatrix} \mathbf{v}(t) \\ \\ \\ \\ \end{pmatrix}$$

- In addition to translating, a rigid body can also spin.



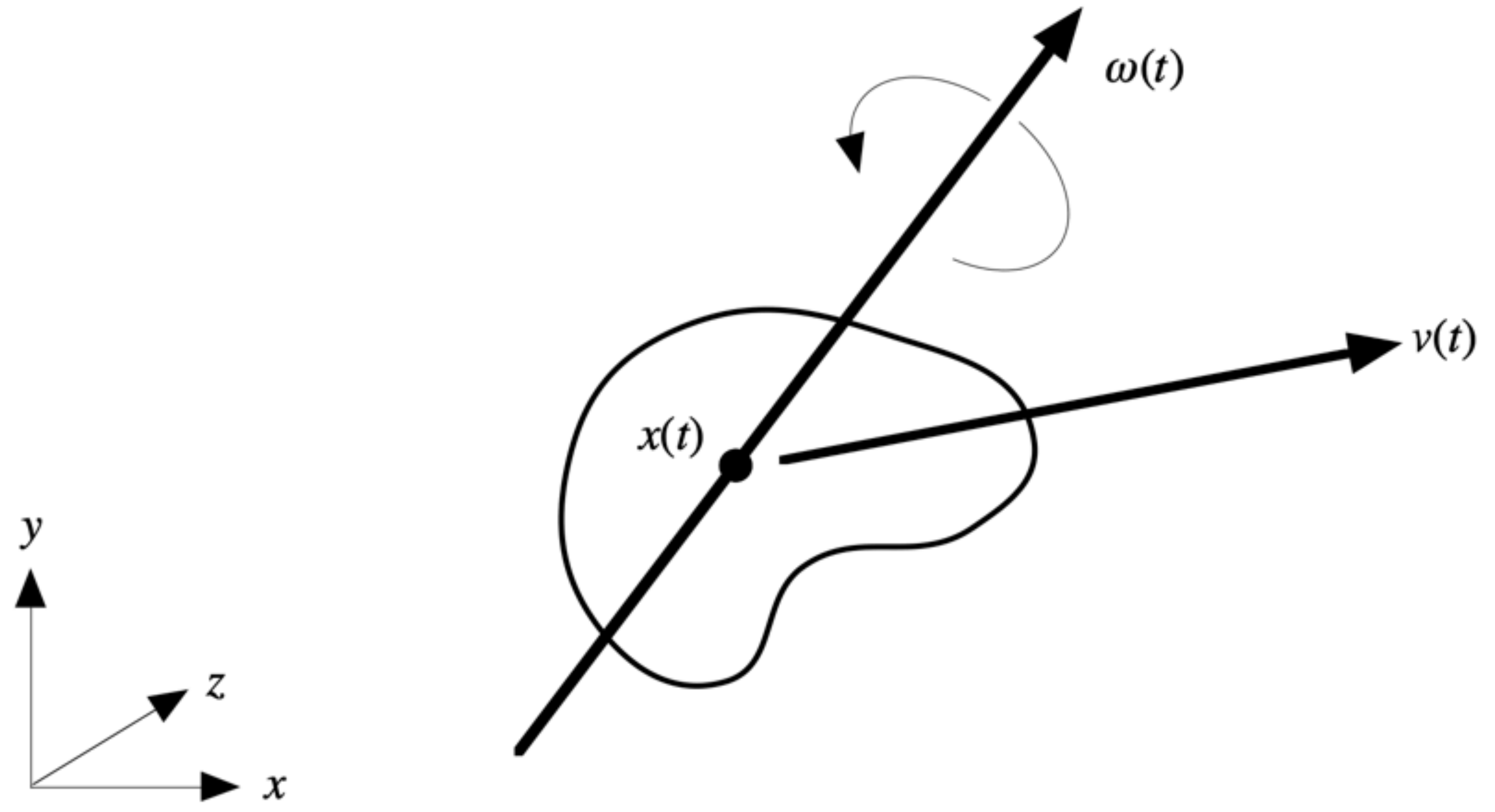
Rigid Body Concepts

Angular Velocity

$$\frac{d}{dt}\mathbf{Y}(t) = \frac{d}{dt}\begin{pmatrix} x(t) \\ R(t) \\ P(t) \\ L(t) \end{pmatrix} = \begin{pmatrix} \mathbf{v}(t) \\ \\ \\ \\ \end{pmatrix}$$

- The magnitude of angular velocity tells how fast the body is spinning.
- How are $R(t)$ and $\omega(t)$ related? (Clearly, the derivative of $R(t)$ cannot be $\omega(t)$, since $R(t)$ is a matrix, and $\omega(t)$ is a vector.)

.

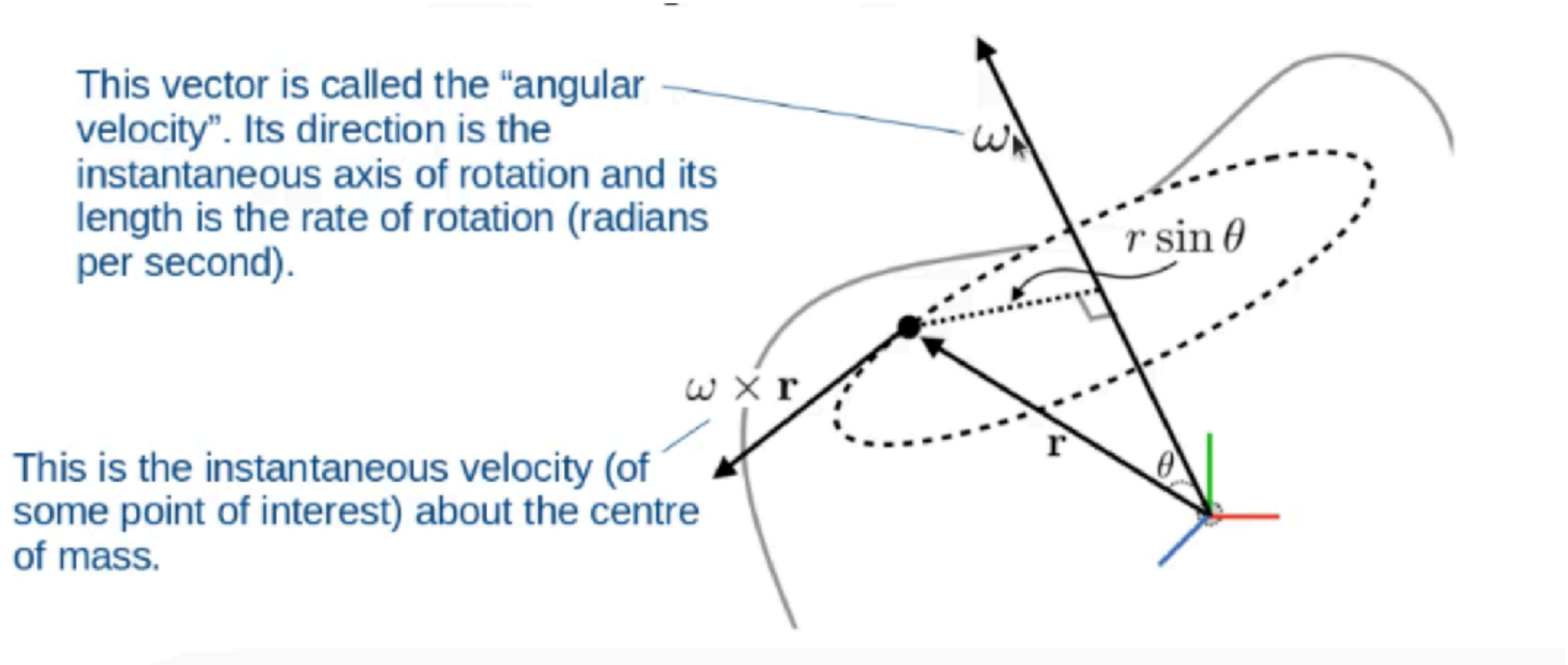


Rigid Body Concepts

Angular Velocity

$$\frac{d}{dt}\mathbf{Y}(t) = \frac{d}{dt}\begin{pmatrix} x(t) \\ R(t) \\ P(t) \\ L(t) \end{pmatrix} = \begin{pmatrix} \mathbf{v}(t) \\ \\ \\ \\ \end{pmatrix}$$

- Let's consider an arbitrary point or particle in the rigid body.
- The instantaneous velocity (the derivative of this point position) of the given point can be computed by $\omega(t) \times r$.

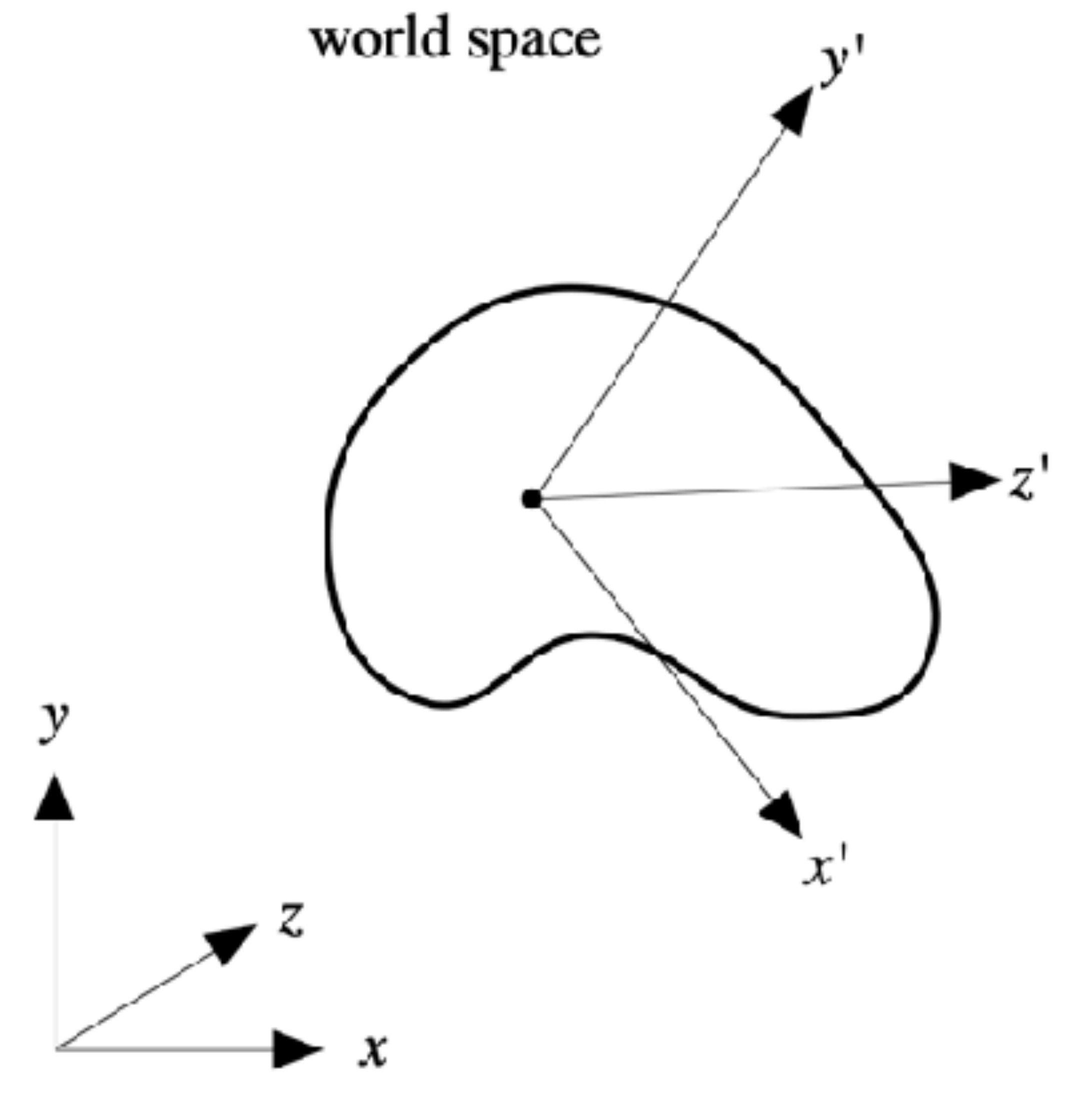


Rigid Body Concepts
Angular Velocity

$$\frac{d}{dt}\mathbf{Y}(t) = \frac{d}{dt}\begin{pmatrix} x(t) \\ R(t) \\ P(t) \\ L(t) \end{pmatrix} = \begin{pmatrix} \mathbf{v}(t) \\ \\ \\ \\ \end{pmatrix}$$

- Physical interpretation of the rotation matrix $R(t)$:
  - Each column of $R(t)$ is the world-space directions that the body-space x, y, and z axes transform to.

$$R(t) = [\, x' \;\; y' \;\; z' \,]$$



Rigid Body Concepts

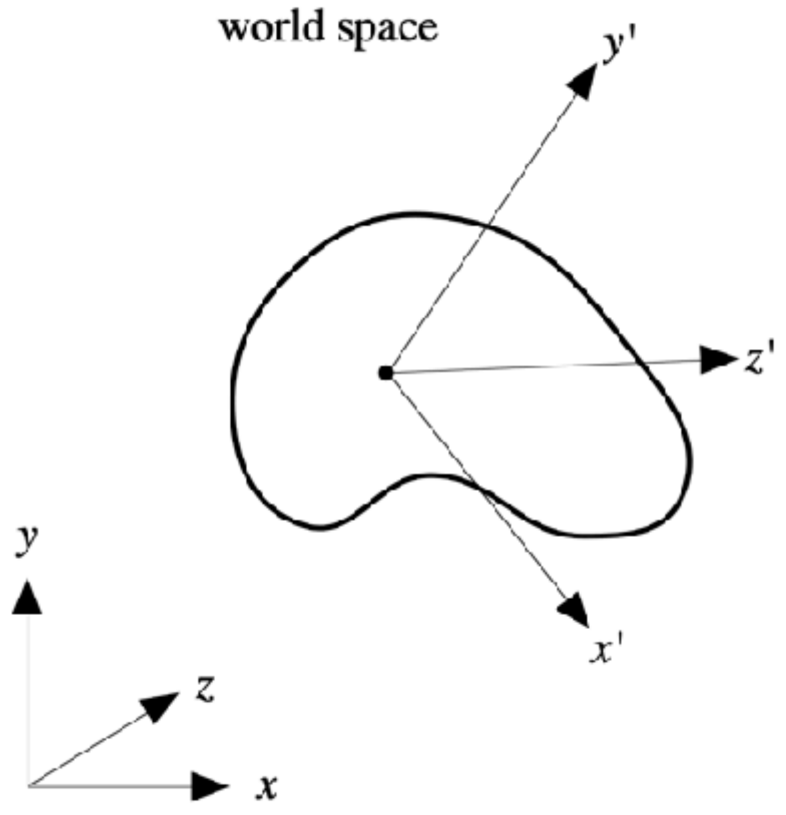
Angular Velocity

$$\frac{d}{dt}\mathbf{Y}(t) = \frac{d}{dt}\begin{pmatrix} x(t) \\ R(t) \\ P(t) \\ L(t) \end{pmatrix} = \begin{pmatrix} \mathbf{v}(t) \\ \\ \\ \\ \end{pmatrix}$$

- Derivative of $R(t)$ is the derivative of positions of three tips x', y', z'.

$$\dot{R} = \left(\omega(t) \times \begin{pmatrix} r_{xx} \\ r_{xy} \\ r_{xz} \end{pmatrix} \quad \omega(t) \times \begin{pmatrix} r_{yx} \\ r_{yy} \\ r_{yz} \end{pmatrix} \quad \omega(t) \times \begin{pmatrix} r_{zx} \\ r_{zy} \\ r_{zz} \end{pmatrix} \right)$$

$R(t) = [\, x' \;\; y' \;\; z' \,]$



- And can be simplified as
- $\dfrac{dR(t)}{dt} = \omega(t)^* R(t)$
- Where $\omega^* = \begin{pmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ \omega_y & \omega_x & 0 \end{pmatrix}$

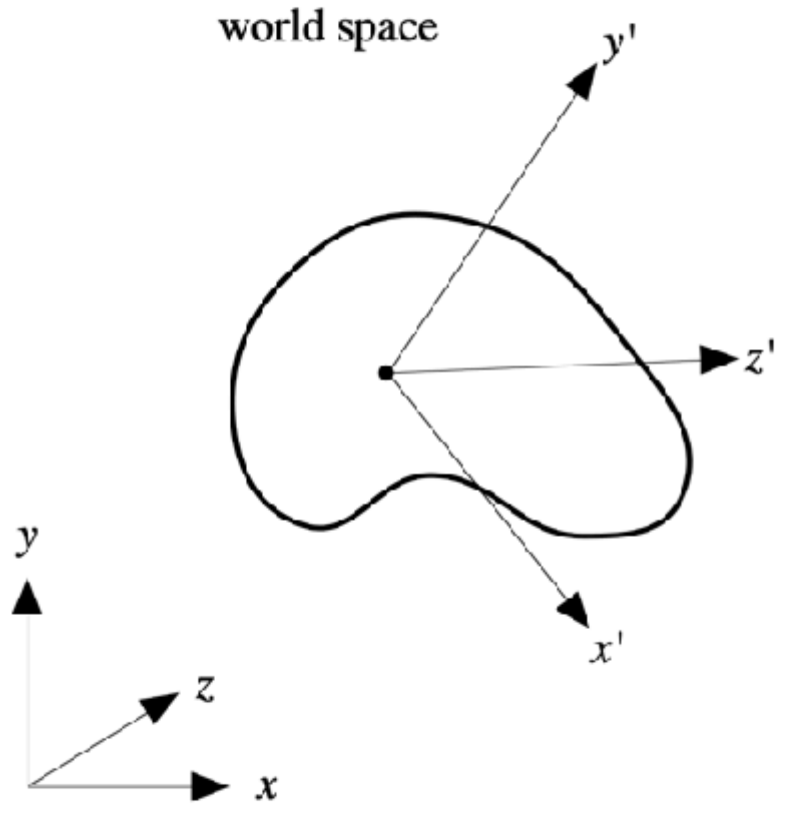
Rigid Body Concepts
Angular Velocity

$$\frac{d}{dt}\mathbf{Y}(t) = \frac{d}{dt}\begin{pmatrix} x(t) \\ R(t) \\ P(t) \\ L(t) \end{pmatrix} = \begin{pmatrix} \mathbf{v}(t) \\ \boldsymbol{\omega}^{\star}(t)\mathbf{R}(t) \\ \\ \\ \end{pmatrix}$$

- Derivative of $R(t)$ is the derivative of positions of three tips x', y', z'.

$$\dot{R} = \left(\omega(t) \times \begin{pmatrix} r_{xx} \\ r_{xy} \\ r_{xz} \end{pmatrix} \quad \omega(t) \times \begin{pmatrix} r_{yx} \\ r_{yy} \\ r_{yz} \end{pmatrix} \quad \omega(t) \times \begin{pmatrix} r_{zx} \\ r_{zy} \\ r_{zz} \end{pmatrix} \right)$$

$R(t) = [\, x' \;\; y' \;\; z' \,]$



- And can be simplified as
- $$\frac{dR(t)}{dt} = \omega(t)^{*}R(t)$$
- Where $\omega^{*} = \begin{pmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ \omega_y & \omega_x & 0 \end{pmatrix}$

Rigid Body Concepts

Force and Linear Momentum

$$\frac{d}{dt}\mathbf{Y}(t) = \frac{d}{dt}\begin{pmatrix} x(t) \\ R(t) \\ P(t) \\ L(t) \end{pmatrix} = \begin{pmatrix} \mathbf{v}(t) \\ \boldsymbol{\omega}^{\star}(t)\mathbf{R}(t) \\ \\ \\ \end{pmatrix}$$

Rigid Body Concepts

Force and Linear Momentum

$$\frac{d}{dt}\mathbf{Y}(t) = \frac{d}{dt}\begin{pmatrix} x(t) \\ R(t) \\ P(t) \\ L(t) \end{pmatrix} = \begin{pmatrix} \mathbf{v}(t) \\ \boldsymbol{\omega}^{\star}(t)\mathbf{R}(t) \\ \\ \\ \end{pmatrix}$$

- $P(t) = Mv(t)$

- $\dfrac{dP(t)}{dt} = M\dfrac{dv(t)}{dt} = Ma(t) = \mathbf{f}(t)$

Rigid Body Concepts

Force and Linear Momentum

$$\frac{d}{dt}\mathbf{Y}(t) = \frac{d}{dt}\begin{pmatrix} x(t) \\ R(t) \\ P(t) \\ L(t) \end{pmatrix} = \begin{pmatrix} \mathbf{v}(t) \\ \boldsymbol{\omega}^{\star}(t)\mathbf{R}(t) \\ \mathbf{f}(t) \\ \end{pmatrix}$$

- $P(t) = Mv(t)$

- $\dfrac{dP(t)}{dt} = M\dfrac{dv(t)}{dt} = Ma(t) = \mathbf{f}(t)$

Rigid Body Concepts

Torque and Angular Momentum

$$\frac{d}{dt}\mathbf{Y}(t) = \frac{d}{dt}\begin{pmatrix} x(t) \\ R(t) \\ P(t) \\ L(t) \end{pmatrix} = \begin{pmatrix} \mathbf{v}(t) \\ \boldsymbol{\omega}^{\star}(t)\mathbf{R}(t) \\ \mathbf{f}(t) \\ \end{pmatrix}$$

- Definition: Angular Momentum of the rigid body is given by the sum of the angular momenta of its constituent particles.

-
$$\begin{aligned} L(t) &= \sum r_i'(t) \times m_i v_i(t) \\ &= \sum m_i r_i'(t) \times (x'(t) + \omega(t) \times r_i'(t)) \\ &= \sum m_i r_i'(t) \times x'(t) + \sum m_i r_i'(t) \times \omega(t) \times r_i'(t) \end{aligned}$$

$$r_i'(t) = r_i(t) - x(t)$$

$$v(t) = x'(t) + \omega(t) \times r_i'(t)$$

Rigid Body Concepts

Torque and Angular Momentum

$$\frac{d}{dt}\mathbf{Y}(t) = \frac{d}{dt}\begin{pmatrix} x(t) \\ R(t) \\ P(t) \\ L(t) \end{pmatrix} = \begin{pmatrix} \mathbf{v}(t) \\ \omega^{\star}(t)\mathbf{R}(t) \\ \mathbf{f}(t) \\ \end{pmatrix}$$

- Definition: Angular Momentum of the rigid body is given by the sum of the angular momenta of its constituent particles.

$$\begin{aligned} L(t) &= \sum r_i'(t) \times m_i v_i(t) \\ &= \sum m_i r_i'(t) \times (x'(t) + \omega(t) \times r_i'(t)) \\ &= \cancel{\sum m_i r_i'(t) \times x'(t)} + \sum m_i r_i'(t) \times \omega(t) \times r_i'(t) \end{aligned}$$

- The first term represents the translation contribution to the angular momentum(zero), and the second represents the rotation contribution.

Rigid Body Concepts

Torque and Angular Momentum

$$\frac{d}{dt}\mathbf{Y}(t) = \frac{d}{dt}\begin{pmatrix} x(t) \\ R(t) \\ P(t) \\ L(t) \end{pmatrix} = \begin{pmatrix} \mathbf{v}(t) \\ \boldsymbol{\omega}^{\star}(t)\mathbf{R}(t) \\ \mathbf{f}(t) \\ \end{pmatrix}$$

- $L(t) = \sum m_i r_i'(t) \times \omega(t) \times r_i'(t)$
- After rearranging
- $$L(t) = \sum m_i r_i'^{*}(t) r_i'^{*T}(t)\omega(t)$$
  $$= I(t)\omega(t)$$
- Where $I(t) = \sum m_i r_i'^{*}(t) r_i'^{*T}(t)$

$$-\mathbf{r}^{\star} = \mathbf{r}^{\star T}$$

$$\mathbf{r}^{\star} = \begin{pmatrix} 0 & -r_z & r_y \\ r_z & 0 & -r_x \\ -r_y & r_x & 0 \end{pmatrix}$$

The inertial tensor $I(t)$ describes the distribution of mass in a rigid body

Rigid Body Concepts

Torque and Angular Momentum

$$\frac{d}{dt}\mathbf{Y}(t) = \frac{d}{dt}\begin{pmatrix} x(t) \\ R(t) \\ P(t) \\ L(t) \end{pmatrix} = \begin{pmatrix} \mathbf{v}(t) \\ \boldsymbol{\omega}^\star(t)\mathbf{R}(t) \\ \mathbf{f}(t) \\ \end{pmatrix}$$

- We can also express the inertia tensor as a transformation of the body space inertia tensor.

$$\begin{aligned}
\mathbf{I}(t) &= \sum_{i=1}^{N} m_i \mathbf{r}_i'^\star(t)\mathbf{r}_i'^{\star T}(t) \\
&= \sum_{i=1}^{N} m_i \left(\mathbf{r}_i'^T \mathbf{r}_i' \boldsymbol{\delta} - \mathbf{r}_i' \mathbf{r}_i'^T\right) \\
&= \mathbf{R}(t) \sum_{i=1}^{N} m_i \left(\mathbf{r}_{0i}'^T \mathbf{r}_{0i}' \boldsymbol{\delta} - \mathbf{r}_{0i}' \mathbf{r}_{0i}'^T\right) \mathbf{R}(t)^T \\
&= \mathbf{R}(t)\mathbf{I}_0\mathbf{R}(t)^T.
\end{aligned}$$

$$\mathbf{r}^\star \mathbf{r}^{\star T} = \mathbf{r}^T \mathbf{r} \boldsymbol{\delta} - \mathbf{r}\mathbf{r}^T$$

$$\mathbf{r} = \mathbf{R}\mathbf{r}_0$$

δ here is the Identity matrix

Where, $I_0 = \sum m_i((r_{0i}^T r_{0i})E - r_{0i} r_{0i}^T)$, r_{0i} is the original particle position from the original body centre of mass

Rigid Body Concepts

Torque and Angular Momentum

$$\frac{d}{dt}\mathbf{Y}(t) = \frac{d}{dt}\begin{pmatrix} x(t) \\ R(t) \\ P(t) \\ L(t) \end{pmatrix} = \begin{pmatrix} \mathbf{v}(t) \\ \boldsymbol{\omega}^{\star}(t)\mathbf{R}(t) \\ \mathbf{f}(t) \\ \end{pmatrix}$$

- The inverse of inertia tensor is given by:

$$\begin{aligned} I^{-1}(t) &= (R(t) I_0 R(t)^T)^{-1} \\ &= (R(t)^T)^{-1} I_0^{-1} R(t)^{-1} \\ &= R(t) I_0^{-1} R(t)^T \end{aligned}$$

Rigid Body Concepts

Torque and Angular Momentum

$$\frac{d}{dt}\mathbf{Y}(t) = \frac{d}{dt}\begin{pmatrix} x(t) \\ R(t) \\ P(t) \\ L(t) \end{pmatrix} = \begin{pmatrix} \mathbf{v}(t) \\ \boldsymbol{\omega}^{\star}(t)\mathbf{R}(t) \\ \mathbf{f}(t) \\ \end{pmatrix}$$

- If force $\mathbf{f}$ is applied to the centre of mass, the body responds like a particle with $\mathbf{a} = \mathbf{f}/M$
- If force $\mathbf{f}$ is applied to elsewhere, this may generate a torque as well.

$$\tau = \mathbf{r} \times \mathbf{f}$$

a point $\mathbf{p}$ located at $\mathbf{r}$ from the centre of mass

Rigid Body Concepts

Torque and Angular Momentum

$$\frac{d}{dt}\mathbf{Y}(t) = \frac{d}{dt}\begin{pmatrix} x(t) \\ R(t) \\ P(t) \\ L(t) \end{pmatrix} = \begin{pmatrix} \mathbf{v}(t) \\ \boldsymbol{\omega}^{\star}(t)\mathbf{R}(t) \\ \mathbf{f}(t) \\ \end{pmatrix}$$

- Rate of change of angular momentum is then torque.

$$\frac{d\mathbf{L}(t)}{dt} = \tau$$

Rigid Body Concepts

Torque and Angular Momentum

$$\frac{d}{dt}\mathbf{Y}(t) = \frac{d}{dt}\begin{pmatrix} x(t) \\ R(t) \\ P(t) \\ L(t) \end{pmatrix} = \begin{pmatrix} \mathbf{v}(t) \\ \boldsymbol{\omega}^{\star}(t)\mathbf{R}(t) \\ \mathbf{f}(t) \\ \boldsymbol{\tau}(t) \end{pmatrix}$$

- Rate of change of angular momentum is then torque.

$$\frac{d\mathbf{L}(t)}{dt} = \tau$$

Summary

$$\frac{d}{dt}\mathbf{Y}(t) = \frac{d}{dt}\begin{pmatrix} x(t) \\ R(t) \\ P(t) \\ L(t) \end{pmatrix}$$

- We have covered all the concepts we need to define the state of a rigid body.
- Centre of mass position $x(t)$
- Orientation $R(t)$
- Linear momentum $P(t) = Mv(t)$
- Angular momentum $L(t) = I(t)\omega(t)$

Summary

- The mass M of the body and body-space inertia tensor I_0 are constants, which we assume we know when the simulation begins
- If we define the following auxiliary quantities

$$v(t) = P(t)/M \qquad I(t) = R(t)I_0R(t)^T \qquad \omega(t) = I(t)^{-1}L(t)$$

- Then the update to the state of the rigid body is given by

$$\frac{d}{dt}\mathbf{Y}(t) = \frac{d}{dt}\begin{pmatrix} x(t) \\ R(t) \\ P(t) \\ L(t) \end{pmatrix} = \begin{pmatrix} \mathbf{v}(t) \\ \boldsymbol{\omega}^{\star}(t)\mathbf{R}(t) \\ \mathbf{f}(t) \\ \boldsymbol{\tau}(t) \end{pmatrix}$$

Implementation

Initialization

Initial conditions of your simulation

$$\mathbf{x}_{cm}, \mathbf{v}_{cm}, \mathbf{R}, \mathbf{L}$$

$$\mathbf{I}^{-1} = \mathbf{R}\mathbf{I}_0^{-1}\mathbf{R}^T$$

$$\omega = \mathbf{I}^{-1}\mathbf{L}$$

Implementation

Forces and torques

$$\tau = \sum_i \mathbf{r}_i \times \mathbf{f}_i$$

$$\mathbf{F} = \sum_i \mathbf{f}_i$$

Implementation

Rigid body state update

$$
\begin{aligned}
\mathbf{x}_{cm} &= \mathbf{x}_{cm} + \Delta t \cdot \mathbf{v}_{cm} \\
\mathbf{v}_{cm} &= \mathbf{v}_{cm} + \Delta t \cdot \mathbf{F}/M \\
\mathbf{R} &= \mathbf{R} + \Delta t \cdot \omega^{\star}\mathbf{R} \\
\mathbf{L} &= \mathbf{L} + \Delta t \cdot \tau \\
\mathbf{I}^{-1} &= \mathbf{R}\mathbf{I}_0^{-1}\mathbf{R}^T \\
\omega &= \mathbf{I}^{-1}\mathbf{L}
\end{aligned}
$$

Implementation

Particle update

$$\mathbf{r}_i = \mathbf{R} \cdot \mathbf{r}_{0i}$$
$$\mathbf{x}_i = \mathbf{x}_{cm} + \mathbf{r}_i$$
$$\mathbf{v}_i = \mathbf{v}_{cm} + \omega \mathbf{r}_i$$

Implementation

Rotation Issue

$$\mathbf{R} = \mathbf{R} + \Delta t \cdot \omega^{\star}\mathbf{R}$$

- Errors accumulate (numerical drift)
  - The matrix becomes no longer orthogonal over time.

Implementation

Rotation Issue solution

- Use quaternion!
- The drifting problem can be alleviated by representing rotations with unit quaternions.
- Any quaternion of unit length corresponds to a rotation, so quaternions deviate from representing rotations only if they lose their unit length. Just normalize it occasionally to fix it.

Implementation

Rotation Issue solution

- Use state $q(t)$ to replace $R(t)$
- Use $\frac{dq(t)}{dt} = \frac{1}{2}[0,\omega_x(t), \omega_y(t), \omega_z(t)]q(t)$ to replace $\frac{dR(t)}{dt} = \omega(t)*R(t)$ when updating the state.

- Note that the rotation matrix is still useful when updating the inertia tensor.
    - Given a quaternion $[s, v_x, v_y, v_z]$, the rotation matrix is given by

$$\begin{pmatrix} 1-2v_y^2-2v_z^2 & 2v_xv_y-2sv_z & 2v_xv_z+2sv_y \\ 2v_xv_y+2sv_z & 1-2v_x^2-2v_z^2 & 2v_yv_z-2sv_x \\ 2v_xv_z-2sv_y & 2v_yv_z+2sv_x & 1-2v_x^2-2v_y^2 \end{pmatrix}$$

Reference

- An Introduction to Physically Based Modeling: Rigid Body Simulation I—Unconstrained Rigid Body Dynamics. David Baraff:
- https://www.cs.cmu.edu/~baraff/sigcourse/notesd1.pdf

Assignment

Due: April 8th 23:59

- Implement a rigid body simulator using Python Taichi Library (90%)
  - 7 TODOs in rigid_body_dynamic.py
  - A recording of your simulation
- Report (10%)
- Bonus: implement extra features or improve the simulation (should introduce details in the report)
- More details will be updated on Github