Game Physics

GPR350, Fall 2019
Daniel S. Buckstein

Angular Dynamics
Week 9

License

Angular Dynamics

- Force, torque and Newton's 2^{nd} law
- Conversion between force and torque
- Moment of inertia
  - Inertia tensor
  - Frame of reference
  - Food for thought

Cool stuff:
http://physics-help.info/physicsguide/appendices/si_units.shtml
https://people.eecs.berkeley.edu/~jfc/mirtich/massProps.html

Recap: Linear Motion

- Position of a particle is determined by integration (e.g. explicit Euler method):

$$x_{t+dt} = x_t + \frac{dx}{dt}dt = x_t + v_t dt$$

$$v_{t+dt} = v_t + \frac{dv}{dt}dt = v_t + a_t dt$$

  - x: position; dx/dt: derivative of position
  - v: velocity; dv/dt: derivative of velocity (acceleration)

Recap: Linear Motion

- Acceleration is determined by rearranging Newton's 2nd law:

$$F = ma$$

$$a = m^{-1}F = \frac{F}{m}$$

  - m: mass (kg)
  - a: acceleration (m/s^2)
  - F: net force on particle (Newtons: N = kg m/s^2)

Recap: Linear Motion

- Force is also related to *linear momentum*:

$$p = mv$$

$$\frac{dp}{dt} = \frac{d}{dt}(mv) = m\frac{dv}{dt}$$

$$= \boldsymbol{ma} = \boldsymbol{F}$$

  - m: mass (kg, ***assuming constant***)
  - v: linear velocity (m/s)
  - p: *linear momentum* (kg m/s)

Angular Motion

- Rotation/orientation is solved using similar methods:

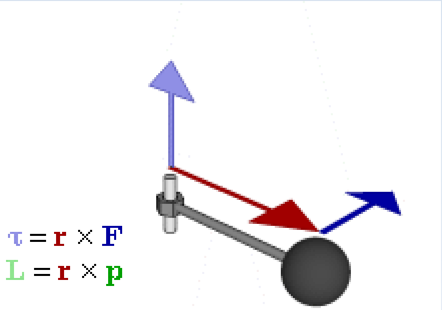
$$q_{t+dt} = q_t + \frac{dq}{dt}dt = q_t + \omega_t q_t \frac{1}{2} dt$$

$$\omega_{t+dt} = \omega_t + \frac{d\omega}{dt}dt = \omega_t + \alpha_t dt$$

  - q: orientation (quaternion)
  - omega: angular velocity; alpha: angular acceleration

Angular Motion

- How do we calculate angular acceleration???
- ***Torque***: "twist force"



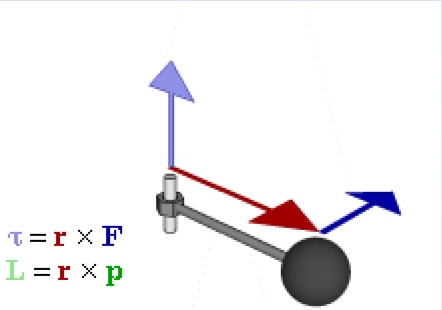
$$\tau = r \times F$$

- r: "moment arm" or "lever arm" vector (m)
- F: force (N)
- tau: *torque* (Newton-meters: Nm)

https://en.wikipedia.org/wiki/Torque#/media/File:Torque_animation.gif

Angular Motion

- How do we calculate angular acceleration???
- ***Angular momentum*** to start with:



$$L = r \times p$$

- r: "moment arm" or "lever arm" vector (m)
- p: linear momentum (kg m/s)
- L: *angular momentum* (kg m^2/s)

https://en.wikipedia.org/wiki/Torque#/media/File:Torque_animation.gif

Angular Motion

- How do we calculate angular acceleration???
- ***Angular momentum***: also calculated with respect to angular velocity:

$$L = I\omega$$

  - I: “moment of inertia” (kg m^2)
  - omega: angular velocity (angle/s)
  - L: *angular momentum* (kg m^2/s)

Angular Motion

- Is torque related to angular momentum?

$$L = I\omega$$

$$\frac{dL}{dt} = \frac{d}{dt}(I\omega) = I\frac{d\omega}{dt}$$

$$= \boldsymbol{I\alpha} = \boldsymbol{\tau}$$

  - I: moment of inertia (kg m^2, ***assuming constant***)
  - alpha: angular acceleration (rad/s^2)
  - tau: net torque on particle (Nm = kg m/s^2)

Angular Motion

- Angular acceleration is determined using the rotational version of Newton's 2nd law:
  - Linear:

$$F = ma$$
$$a = m^{-1}F$$

  - Angular:

$$\tau = I\alpha$$
$$\alpha = I^{-1}\tau$$

Angular Motion

- Can convert between linear and angular model using the original torque formula
  - Linear to angular:

$$\tau = r \times F$$

  - Angular to linear:

$$F = \tau \times r$$

Moment of Inertia

- Problem: in vector physics, ***moment of inertia is not necessarily just a scalar***…
- Effectively represents "resistance to angular change over the distribution of mass"
- Mass as a *uniform scale* rather than a *scalar*:

$$\vec{F} = \boldsymbol{m}_{3\times 3}\, \vec{a} = \begin{bmatrix} m & & \\ & m & \\ & & m \end{bmatrix} \begin{bmatrix} \vec{a}_x \\ \vec{a}_y \\ \vec{a}_z \end{bmatrix}$$

Moment of Inertia

- Recovering acceleration from force would use the inverse of this uniform scale:

$$\vec{a} = \boldsymbol{m}_{3\times3}^{-1}\vec{F} = \begin{bmatrix} m^{-1} & & \\ & m^{-1} & \\ & & m^{-1} \end{bmatrix}\vec{F}$$

- We might call this a “mass tensor”

Moment of Inertia

- This rule is used for angular since inertia is not necessarily uniform across a rigid body:

$$\vec{\tau} = \boldsymbol{I}_{3\times 3}\,\vec{\alpha}$$
$$\vec{\alpha} = \boldsymbol{I}_{3\times 3}^{-1}\,\vec{\tau}$$

- The moment of inertia in matrix form is called an "***inertia tensor***"
- Pre-defined for some shapes, or calculate in general form

Inertia Tensor

- Examples (uniform distribution):
  - Solid sphere of radius *r* and mass *m*

$$I = \begin{bmatrix} \frac{2}{5}mr^2 & & \\ & \frac{2}{5}mr^2 & \\ & & \frac{2}{5}mr^2 \end{bmatrix}$$

  - Hollow sphere of radius *r* and mass *m*

$$I = \begin{bmatrix} \frac{2}{3}mr^2 & & \\ & \frac{2}{3}mr^2 & \\ & & \frac{2}{3}mr^2 \end{bmatrix}$$

http://dynref.engr.illinois.edu/rem.html
https://en.wikipedia.org/wiki/List_of_moments_of_inertia

Inertia Tensor

- Examples (non-uniform distribution):
  - Solid box* of width *w*, height *h,* depth *d* and mass *m*

$$I = \begin{bmatrix} \frac{1}{12}m(h^2+d^2) & & \\ & \frac{1}{12}m(d^2+w^2) & \\ & & \frac{1}{12}m(w^2+h^2) \end{bmatrix}$$

  - Hollow box** of width *w*, height *h,* depth *d* and mass *m*

$$I = \begin{bmatrix} \frac{5}{3}m(h^2+d^2) & & \\ & \frac{5}{3}m(d^2+w^2) & \\ & & \frac{5}{3}m(w^2+h^2) \end{bmatrix}$$

https://physics.stackexchange.com/questions/105229/tensor-of-inertia-of-a-hollow-cube/105234
https://en.wikipedia.org/wiki/List_of_moments_of_inertia

*cube: $I = \frac{1}{6}ms^2$

**inferred

Inertia Tensor

- Examples (non-uniform distribution):
  - Solid cylinder* of radius *r*, height *h* and mass *m*

$$I = \begin{bmatrix} \frac{1}{12}m(3r^2 + h^2) & & \\ & \frac{1}{12}m(3r^2 + h^2) & \\ & & \frac{1}{2}mr^2 \end{bmatrix}$$

  - Solid cone* of radius *r*, height *h* and mass *m* about apex

$$I = \begin{bmatrix} \frac{3}{5}mh^2 + \frac{3}{20}mr^2 & & \\ & \frac{3}{5}mh^2 + \frac{3}{20}mr^2 & \\ & & \frac{3}{10}mr^2 \end{bmatrix}$$

*axis parallel to third local basis

https://en.wikipedia.org/wiki/List_of_moments_of_inertia

Inertia Tensor

- Beware of rods: *inverse cannot be "real"* !
  - Rod* of length *l* and mass *m* spinning about end

$$I = \begin{bmatrix} \frac{1}{3}ml^2 & & \\ & \frac{1}{3}ml^2 & \\ & & 0 \end{bmatrix}$$

nope ☹

  - Rod* of length *l* and mass *m* spinning about center

$$I = \begin{bmatrix} \frac{1}{12}ml^2 & & \\ & \frac{1}{12}ml^2 & \\ & & 0 \end{bmatrix}$$

*axis parallel to third local basis

https://en.wikipedia.org/wiki/List_of_moments_of_inertia

Inertia Tensor

- General tensor for a *rigid system of particles*:
- For a list of n particles, each with mass m_k and position x_k, calculate the total mass M and center of mass c_m:

$$M = \sum_{k=0}^{n-1} m_k$$

$$c_m = \frac{1}{M} \sum_{k=0}^{n-1} m_k x_k$$

Moment arm can be calculated at any point:

$$r = x - c_m$$

Center of mass c_m is our "reference point"

Inertia Tensor

- Inertia tensor from a set of masses:

$$I = \begin{bmatrix} I_{xx} & I_{yx} & I_{zx} \\ I_{xy} & I_{yy} & I_{zy} \\ I_{xz} & I_{yz} & I_{zz} \end{bmatrix}$$

$$I_{xy} = I_{yx}, \qquad I_{yz} = I_{zy}, \qquad I_{zx} = I_{xz}$$

$$r_k = x_k - c_m = (a_k, b_k, c_k)$$

Inertia Tensor

- Calculate the terms:

$$I_{xx} = \sum_{k=0}^{n-1} m_k \left(b_k^2 + c_k^2\right)$$

$$I_{xy} = -\sum_{k=0}^{n-1} m_k a_k b_k$$

$$I_{yy} = \sum_{k=0}^{n-1} m_k \left(c_k^2 + a_k^2\right)$$

$$I_{yz} = -\sum_{k=0}^{n-1} m_k b_k c_k$$

$$I_{zz} = \sum_{k=0}^{n-1} m_k \left(a_k^2 + b_k^2\right)$$

$$I_{zx} = -\sum_{k=0}^{n-1} m_k c_k a_k$$

Inertia Tensor

- The inertia tensor is inverted and used to *solve angular acceleration*:

$$\vec{\tau} = I\vec{\alpha}$$

$$\vec{\alpha} = I^{-1}\vec{\tau}$$

- The final problem: ***frame of reference***
- The vectors are in *world space*, while the tensor (and its inverse) is *local* to the object…

$$\vec{\alpha}_{\text{world}} = I_{\text{local}}^{-1}\vec{\tau}_{\text{world}}$$

nope ☹

Inertia Tensor

- Perform ***change of basis***: bring torque into local space, apply tensor, move back to world
- ${}^{\text{world}}R_{\text{local}}$ is the object's *world* orientation:

$$\vec{\alpha}_{\text{world}} = \left({}^{\text{world}}R_{\text{local}}\, I_{\text{local}}^{-1}\, {}^{\text{local}}R_{\text{world}}\right)\vec{\tau}_{\text{world}}$$

$$\vec{\alpha}_{\text{world}} = \left({}^{\text{world}}R_{\text{local}}\, I_{\text{local}}^{-1}\, {}^{\text{world}}R_{\text{local}}^{-1}\right)\vec{\tau}_{\text{world}}$$

Inertia Tensor

- Perform ***change of basis***: bring torque into local space, apply tensor, move back to world
- Therefore, every update we calculate a "modified" inertia tensor (inverse) to use:

$$I_t^{-1} = R\, I^{-1}\, R^{-1}$$

- The final angular acceleration update:

$$\alpha_t = I_t^{-1} \tau_t$$

Inertia Tensor

- ***Pro tip***: know the behavior of uniform scales:

$$S = \begin{bmatrix} s & & \\ & s & \\ & & s \end{bmatrix}$$

- What happens when you multiply it with any other 3x3 matrix '*A*'?

$$SA \equiv sA = As \equiv AS$$

- A uniform scale matrix commutes!

Inertia Tensor

- *If your inertia tensor is a uniform scale* (e.g. sphere, cube), can cancel out change of basis:

$$I_t^{-1} = R\, I^{-1}\, R^{-1}$$
$$\equiv \color{red}{R\, R^{-1}}\, I^{-1}$$

- Technically this also happens for mass…

$$a = R\, \boldsymbol{m}_{3\times 3}^{-1}\, R^{-1}\, F$$
$$a \equiv \color{red}{R\, R^{-1}}\, \boldsymbol{m}_{3\times 3}^{-1}\, F$$

$$a = m_t^{-1} F \qquad m_t^{-1} = m^{-1}$$ (constant scalar)

Inertia Tensor

- Force application will also be off if we do not also update the center of mass (which is local)

$$r_{\mathrm{f}_{\mathrm{world}}} = x_{\mathrm{f}_{\mathrm{world}}} - c_{m_{\mathrm{local}}}$$ nope ☹

$$c_{m_{\mathrm{world}}} = {}^{\mathrm{world}}R_{\mathrm{local}}\, c_{m_{\mathrm{local}}}$$

- All operations now solvable in world space:

$$r_{\mathrm{f}_{\mathrm{world}}} = x_{\mathrm{f}_{\mathrm{world}}} - c_{m_{\mathrm{world}}}$$
$$\tau_{\mathrm{world}} = r_{\mathrm{f}_{\mathrm{world}}} \times F_{\mathrm{world}}$$
$$\alpha_{\mathrm{world}} = I_{\mathrm{world}}^{-1}\, \tau_{\mathrm{world}}$$

Rigid Body Dynamics

- We now have a complete set of dynamics formulas in the same frame of reference:

Force generators:

$$r_{\mathrm{f}} = x_{\mathrm{f}} - c_m$$
$$\tau = r_{\mathrm{f}} \times F$$

$$F_t = \sum F$$
$$\tau_t = \sum \tau$$

$$a_t = m_t^{-1} F_t = m^{-1} F_t$$
$$\alpha_t = I_t^{-1} \tau_t = R\, I^{-1}\, R^{-1}\, \tau_t$$

Integration (Euler):

$$x_{t+dt} = x_t + v_t dt$$

$$v_{t+dt} = v_t + a_t dt$$

$$q_{t+dt} = q_t + \omega_t q_t \frac{dt}{2}$$
$$q'_{t+dt} = \hat{q}_{t+dt}$$

$$\omega_{t+dt} = \omega_t + \alpha_t dt$$

Rigid Body Dynamics

- Food for thought: relationship with velocity:

$$\omega = \frac{r \times v}{|r|^2}$$

$$I = m|r|^2$$

- Therefore,

$$L = I\omega = m|r|^2 \frac{r \times v}{|r|^2}$$

$$= m(r \times v) = r \times mv = r \times p$$

Rigid Body Dynamics

- Final food for thought:
- Minimize the number of loops
  - E.g. calculate total mass and center of mass simultaneously
  - E.g. calculate and store inertia tensor (and inverse) once per object
- Don’t forget to normalize orientation after integration!

Rigid Body Dynamics

- ***Challenge!***
- Prove the tensor formulas for spheres and cubes using a collection of "particles"
- I.e. read discrete mesh data
- Box: calculation should yield solid or hollow box tensor
- Sphere: higher resolution; should be closer to solid or hollow tensor

The end.

- Questions? Comments? Concerns?