

There are 12 Degrees of Freedom (DOF) in an unconstrained rigid body:

3 for position $q = [q_x, q_y, q_z]$

3 for rotation $\omega = [\omega_x, \omega_y, \omega_z]$

3 for linear velocity \dot{q}

3 for angular velocity $\dot{\omega}$

But - computationally speaking - its rare that the state vector is represented as $[q, \omega, \dot{q}, \dot{\omega}]$. More commonly Linear Momentum P and Angular Momentum. L replace their respective velocities. So our state vector is now $[q, \omega, P, L]$.

Linear Momentum $P(t) : P(t) = m \cdot v(t)$

m is the mass

$v(t)$ is the linear velocity

Angular Momentum $L(t) : L(t) = I(t) \cdot \omega(t)$

$I(t)$ is the moment of inertia - also referred to as the inertia tensor - explained later.

$\omega(t)$ is the angular velocity

Notice the similarity of the two. Also notice that unlike mass - the moment of inertia is changes with time.

Understanding the motion of the rigid body.

Unlike point-masses - we have additional problems to consider.

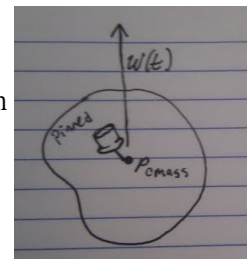
Rigid bodies collide with each other with points away from their center of mass.

- 1.) If the object is spinning - the point of collision has a different velocity than the center of mass.
- 2.) A collision will act not only on the velocity of the center of mass of the object - but will also apply a torque to it which is dependent on the handle.

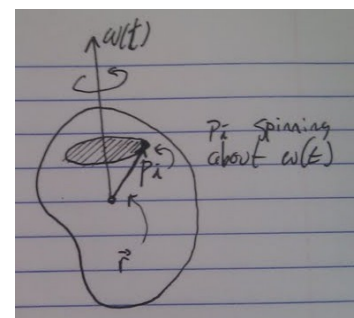
So to simulate both 1 and 2 properly we need to understand the rotational component of the rigid body.

A rigid body's movement can be broken down into a linear component $v(t)$ and an angular component $\omega(t)$

How fast is any other point p_i on the body moving?
 Imagine the body is "pinned" (it has no linear velocity) at its center of mass which we will denote by p_{cmass} and rotates about an axis $\omega(t)$. Note that the t implies that this axis of rotation can change with time - picture a spinning top that gradually loses its ability to stay upright.



Create a vector from the p_{cmass} to another point p_i denoted by r_i . Imagine the circular path of p_i as it rotates about $\omega(t)$. The velocity of our point is $v_p = \omega \times r_i$. This should be intuitive - $\omega \times r_i$ gives us a vector orthogonal to r_i and the axis of rotation ω - precisely the direction p_i is moving. One thing that may not be obvious here is that the amount of rotation about ω is given by $|\omega|$. So if ω were longer - the body rotates more about ω . If this were not so - v_p would simply point in the direction that p_i is headed but would not - in general - be the actual velocity of p_i .



Now "unpin" our rigid body. It now has its own linear velocity denoted by v_{cmass} . How is p_i moving now? Simply $v_{p_i} = v_{cmass} + \omega \times r_i$

An aside for some notation: we can represent $\omega \times a$ where a is an arbitrary 3 vector in matrix form like so:

$$\begin{bmatrix} 0 & -\omega_2 & \omega_1 \\ \omega_2 & 0 & -\omega_0 \\ -\omega_1 & \omega_0 & 0 \end{bmatrix} \times \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix}$$

In notation we write this as $\tilde{\omega}$

Some notes on $\tilde{\omega}$

$\tilde{\omega}$ is skew symmetric: $\omega^{-1} = \omega^T$

diagonals have to be 0 since $a_{11} = -a_{11}$ and $a_{22} = -a_{22}$ etc

Notice that we don't need to store p_i explicitly; It can be written in terms of its offset from the center of mass and the objects Xform: $\bar{P}_i = x(t) + R(t) \cdot \bar{P}_{object}$

$x(t)$ is a translation and $R(t)$ is a rotation represented by a 3x3 matrix. The bar above P indicates were dealing with world-space coordinates - not object space ones.

A note about $R(t)$ We can store this as a 3x3 matrix but we would be storing 9 values to represent 3 DOFs. An alternative is to store our objects rotation as a quaternion (4 values for 3 DOFs) and compute our 3x3 rotation matrix from it only when needed.

Okay - were only missing one more thing. The inertia tensor. $I(t)$ is a 3x3 matrix that is the scaling factor between angular momentum and angular velocity. Informally - very informally - it quantifies how easy it is to spin an object as a function of the distribution of mass for that object. Picture the difference between spinning a dense ball of iron surrounded by a large amount of foam, and large ball of foam surrounded by a thin - but massive - layer of iron.

Specifically, its the integral of the mass over the volume. It changes with the rotation of the rigid body - so it varies with t . At first glance it looks as if we have to compute an expensive integral for each time-step in the simulation. But we can do some of the math in body space once for each rigid body and before the simulation starts to reduce the calculation at runtime.

To update I for each time-step we can use take the conjugate with the bodies rotation matrix.

$$I = R(t) \times I_{body} \times R^T(t)$$

So what is I_{body} and how is it computed? I_{body} Is the inertia tensor at our bodies rest state.

$$\text{In a continuous setting } I_{body} = \int_{\Omega} \rho (\|\vec{r}\|^2 I - \vec{r} \vec{r}^T) \delta V$$

In a discrete setting we can represent it like this: $I_{body} = \sum m_i ((r_{0i}^T r_{0i}) 1 - r_{0i} r_{0i}^T)$ and is stored as a 3x3 matrix.

The r here is the displacement of the i'th particle from the objects center, or in simpler terms the the body space coordinate of the i'th particle. m_i is the mass of the i'th particle and the 1 here is the identity matrix.

Now we have everything we need to derive the equations of motion. From our state vector we can derive the following auxiliary quantities:

$$\dot{P} = F \quad \dot{L} = \tau \quad \tau = r \times F \quad v(t) = \frac{P(t)}{M} \quad \omega(t) = I(t)^{-1} L(t)$$