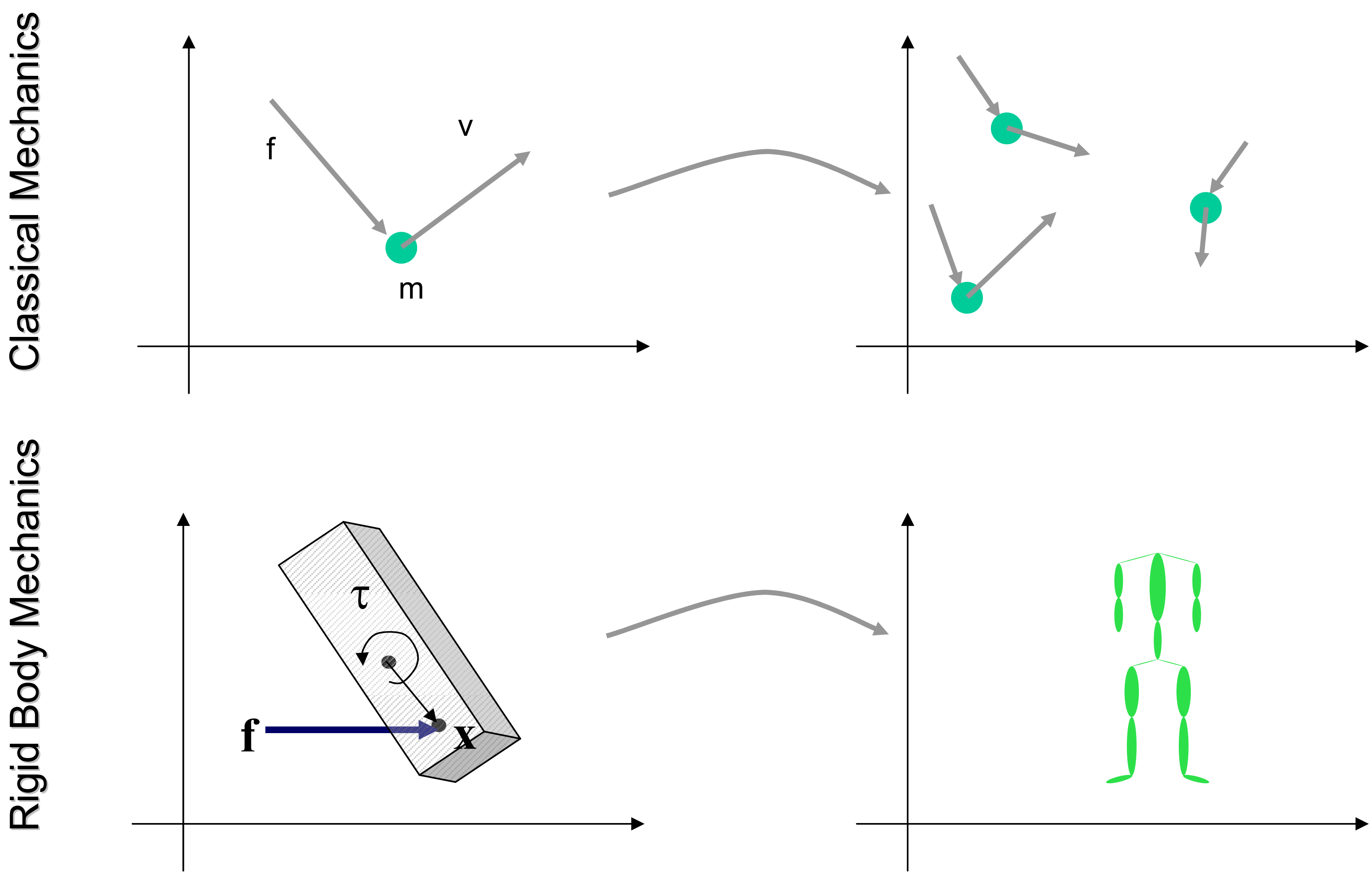
Physics-Based Models for People Tracking: Classical and Constrained Mechanics

Marcus Brubaker[1] Leonid Sigal[1,2] David J Fleet[1]

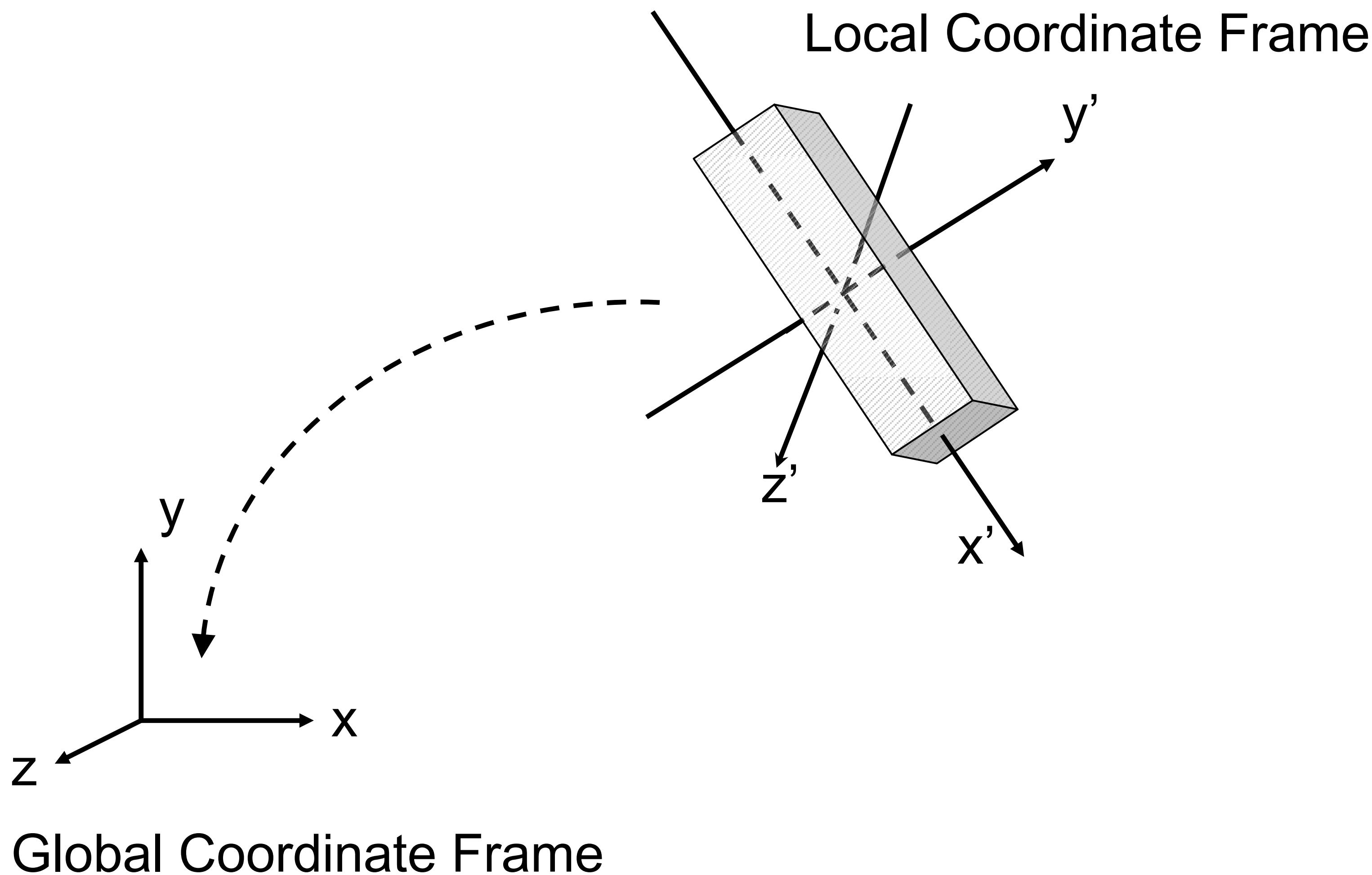
[1]University of Toronto

[2]Disney Research, Pittsburgh

Classical and Rigid Body Mechanics



Pose of a Rigid Body



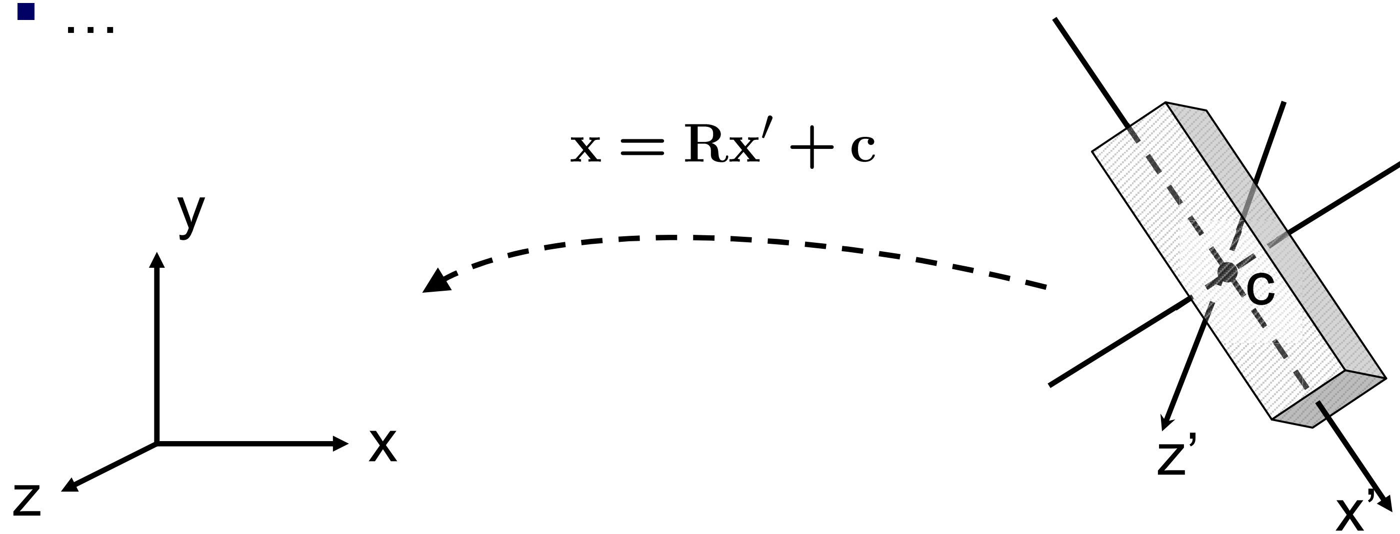
Pose of a Rigid Body

Pose is the rigid transformation from a local coordinate frame to a global coordinate frame

Orientation can be parameterized in a number of ways

- Rotation matrices
- Euler angles
- Exponential maps
- …

$$\mathbf{x} = \mathbf{R}\mathbf{x}' + \mathbf{c}$$



Representing Orientation with Quaternions

Unit vectors in $\mathbb{R}^4$ form a singularity free represent of orientation in 3D

$$\mathbf{q} = \begin{pmatrix} \cos(\theta/2) \\ \sin(\theta/2)\mathbf{n} \end{pmatrix} = \begin{pmatrix} w \\ x \\ y \\ z \end{pmatrix} \qquad \mathbf{x} = \mathbf{R}(\mathbf{q})\mathbf{x}'$$

$$\mathbf{R}(\mathbf{q}) = \begin{pmatrix} w^2+x^2-y^2-z^2 & 2(xy-wz) & 2(xz-wy) \\ 2(yx+wz) & w^2-x^2+y^2-z^2 & 2(yz-wx) \\ 2(zx-wy) & 2(zy+wx) & w^2-x^2-y^2+z^2 \end{pmatrix}$$

Using *quaternion algebra*

quaternion multiplication

quaternion inverse

$$\begin{pmatrix} 0 \\ \mathbf{x} \end{pmatrix} = \mathbf{q} \circ \begin{pmatrix} 0 \\ \mathbf{x}' \end{pmatrix} \circ \mathbf{q}^{-1}$$

Representing Orientation with Quaternions

Quaternion algebra at a glance

$$\mathbf{q}_0 = \begin{pmatrix} w_0 \\ \mathbf{u}_0 \end{pmatrix} \qquad \mathbf{q}_1 = \begin{pmatrix} w_1 \\ \mathbf{u}_1 \end{pmatrix}$$

- Quaternion multiplication (i.e., composition)

$$\mathbf{q}_0 \circ \mathbf{q}_1 = \begin{pmatrix} w_0 w_1 - \mathbf{u}_0 \cdot \mathbf{u}_1 \\ w_0 \mathbf{u}_1 + w_1 \mathbf{u}_0 + \mathbf{u}_0 \times \mathbf{u}_1 \end{pmatrix}$$

- Inverse of a quaternion

$$\mathbf{q}^{-1} = \|\mathbf{q}\|^{-2} \begin{pmatrix} w \\ -\mathbf{u} \end{pmatrix}$$

- Quaternion multiplication as matrix-vector multiplication

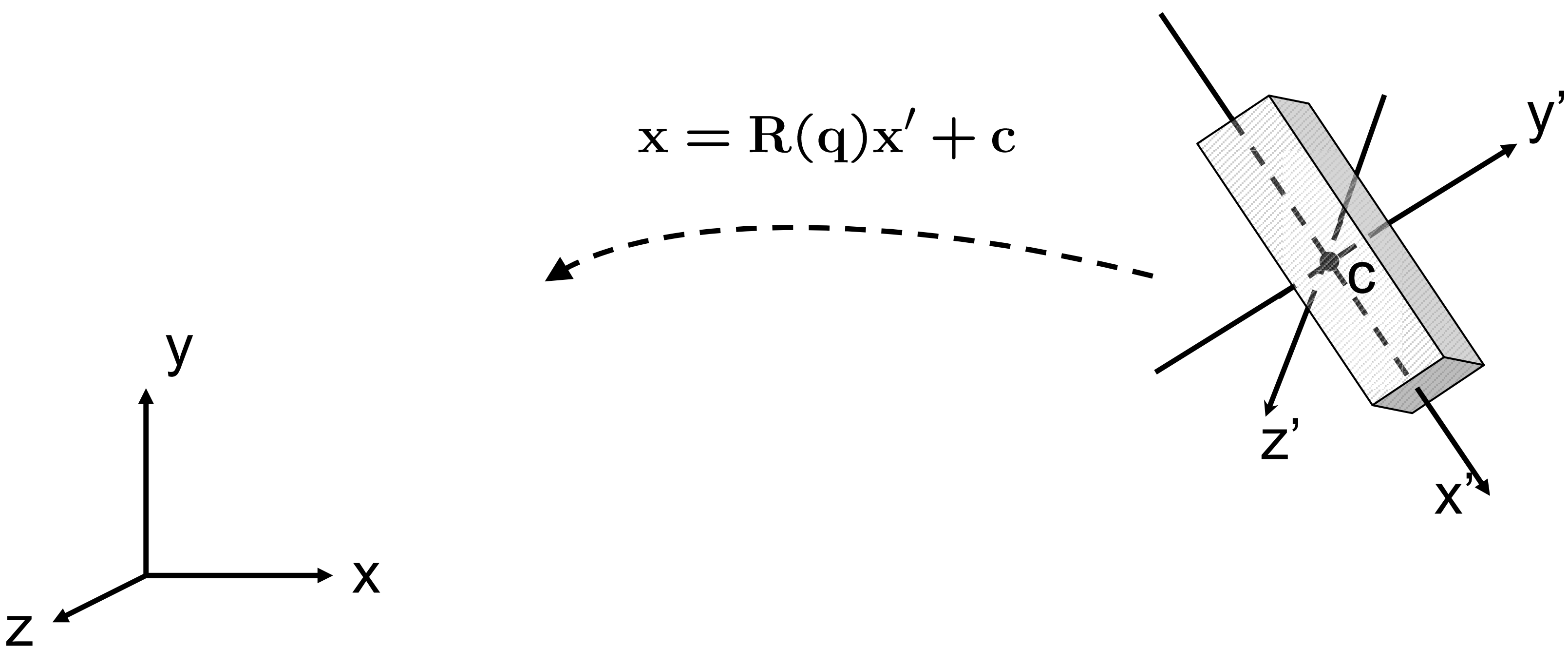
$$\mathbf{q}_0 \circ \mathbf{q}_1 = \mathbf{Q}(\mathbf{q}_0)\mathbf{q}_1 = \bar{\mathbf{Q}}(\mathbf{q}_1)\mathbf{q}_0$$

Pose of a Rigid Body

Pose consists of…

- Position $\mathbf{c}$
- Orientation $\mathbf{q}$

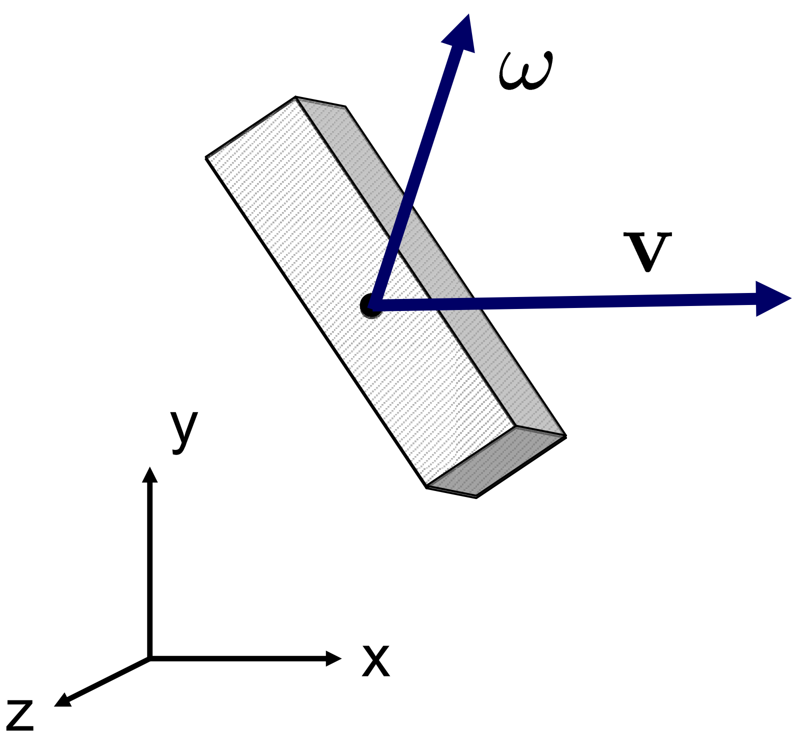
$$\mathbf{x} = \mathbf{R}(\mathbf{q})\mathbf{x}' + \mathbf{c}$$



Rigid-Body Mechanics: Velocity

Mechanics is about *motion*

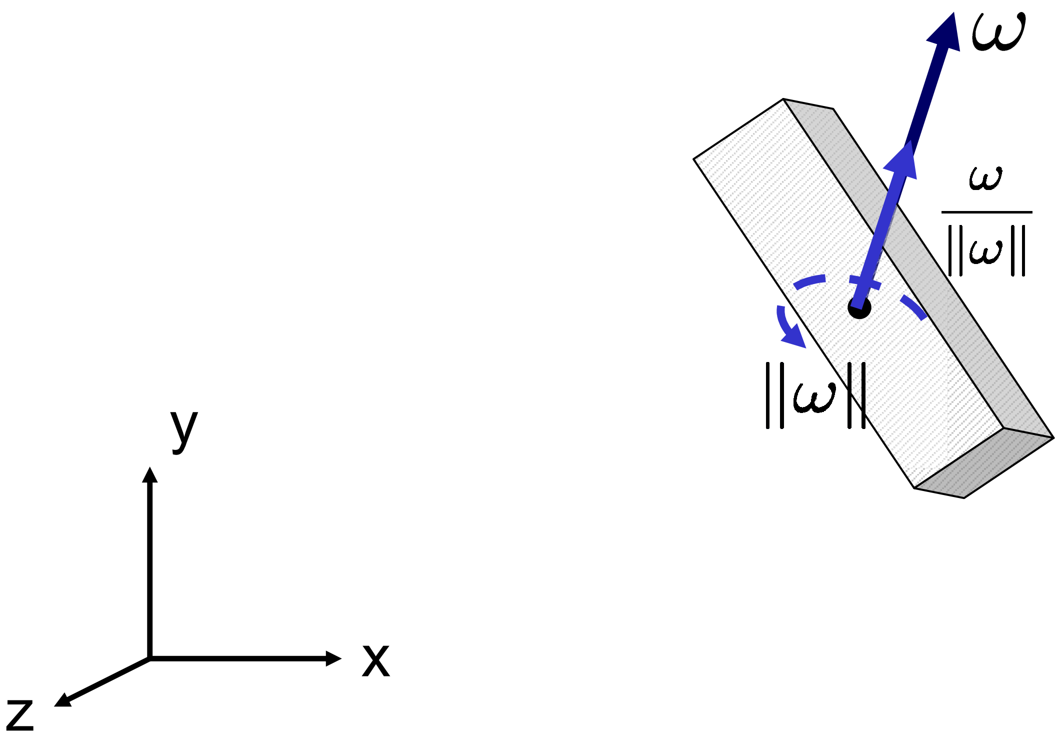
- Linear velocity
- Angular velocity



Rigid-Body Mechanics: Velocity

Angular velocity vector

- Direction is the axis of rotation
- Magnitude is instantaneous rate of rotation



Rigid-Body Mechanics: Mass

- Mass is defined in terms of a *mass density function* $\rho(\mathbf{x})$

Total Mass

$$m = \int \rho(\mathbf{x}) d\mathbf{x}$$

Center of Mass

$$\mathbf{c} = m^{-1} \int \mathbf{x} \rho(\mathbf{x}) d\mathbf{x}$$

Rigid-Body Mechanics: Momentum

Linear momentum

- mass times linear velocity

$$\mathbf{p} = m\mathbf{v}$$

Angular momentum

- same idea, but angular momentum depends on *distribution* of mass
- *inertia tensor* times angular velocity

$$\ell = \mathbf{I}\omega$$

Rigid-Body Mechanics

Inertia Tensor about the Center of Mass

- Real, symmetric matrix summarizing the distribution of mass about the center of mass

$$\mathbf{I} = \begin{pmatrix} I_{11} & I_{12} & I_{13} \\ I_{21} & I_{22} & I_{23} \\ I_{31} & I_{32} & I_{33} \end{pmatrix}$$

- Diagonal entries called the *moments of inertia*, off diagonals call the *products of inertia*
- Computed as a second moment of the mass density function

$$\mathbf{I} = \int \left(\|\mathbf{r}\|^2 \mathbf{E}_{3\times 3} - \mathbf{r}\mathbf{r}^T \right) \rho(\mathbf{x}) d\mathbf{x}$$

$$\mathbf{r} = \mathbf{x} - \mathbf{c}$$

Rigid-Body Mechanics

Inertia Tensor about the Center of Mass

- In a local coordinate frame defined by the eigenvectors the inertia tensor is diagonal

$$\mathbf{I}' = \begin{pmatrix} I_1 & 0 & 0 \\ 0 & I_2 & 0 \\ 0 & 0 & I_3 \end{pmatrix}$$

- Diagonals are the *principal moments of inertia* and the corresponding axes are the *principal axes of inertia*
- When the coordinate system is rotated

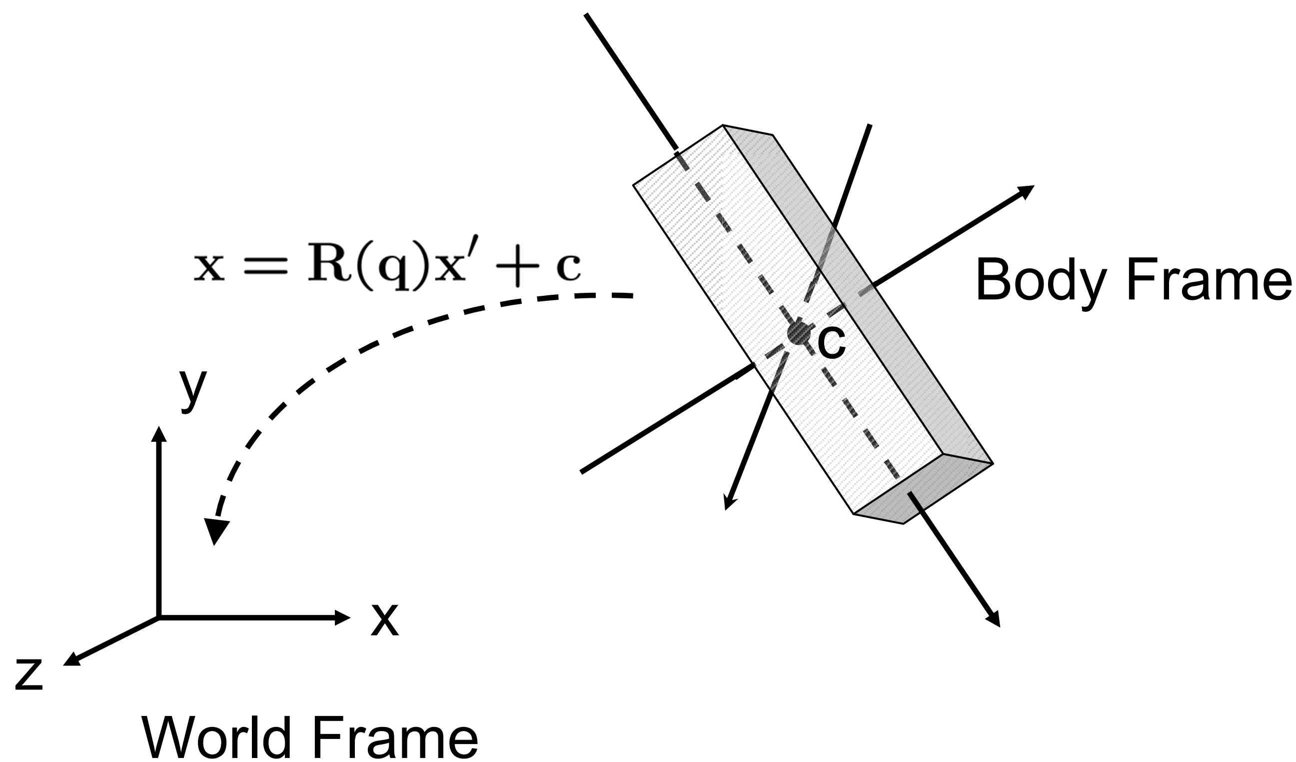
$$\mathbf{I} = \mathbf{R}\mathbf{I}'\mathbf{R}^T$$

- Only need to compute the integral once

Rigid-Body Mechanics

Pose of a Rigid Body

- The pose of a rigid body specifies the rigid transformation from the *body frame* to the *world frame*
- The *body frame* has its origin at the center of mass and its axes aligned with the principal axes of inertia



Rigid-Body Mechanics: Equations of Motion

Newton`s Second Law of Motion

- *In a static frame of reference, the time derivative of* momentum *is force.*

$$\begin{aligned} \mathbf{f} &= \dot{\mathbf{p}} \\ &= m\dot{\mathbf{v}} \end{aligned}$$

$$\begin{aligned} \tau &= \dot{\ell} \\ &= \mathbf{I}\dot{\omega} + \dot{\mathbf{I}}\omega \end{aligned}$$

Inertia tensor is not constant in a static frame

Rigid-Body Mechanics

Frames of Reference

- The Newton-Euler equations are for motion measured in a motionless frame of reference (i.e., the world frame)
- It is useful to consider rotational motion in the body frame

$$\mathbf{I}'\dot{\omega}' = \tau' - \omega' \times (\mathbf{I}'\omega')$$

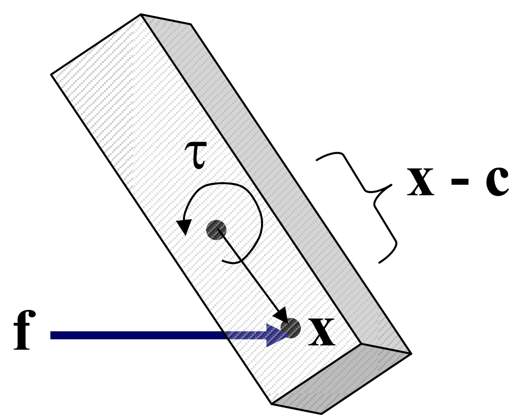
- Since the inertia tensor is constant and diagonal this equation is easier to work with

Rigid-Body Mechanics

Force and Torque

- Forces (torques) in equations of motion are applied on (about) the center of mass of the object
- Force applied at a point results in both a force on the center of mass and a torque about the center of mass

$$\tau = \mathbf{f} \times (\mathbf{x} - c)$$



Rigid-Body Mechanics

Force and Torque

- Forces (torques) in equations of motion are applied on (about) the center of mass of the object
- Force applied at a point results in both a force on the center of mass and a torque about the center of mass
- Force is linear, so the net of all forces on a body can be summarized by a force on the center of mass and a torque about the center of mass

Rigid-Body Mechanics

Relating Angular Velocity to Quaternions

- Angular velocity is independent of representation of orientation

$$\dot{\mathbf{q}} = \frac{1}{2}\begin{pmatrix} 0 \\ \omega \end{pmatrix} \circ \mathbf{q} \qquad \dot{\mathbf{q}} = \frac{1}{2}\mathbf{q} \circ \begin{pmatrix} 0 \\ \omega' \end{pmatrix}$$

Rigid-Body Mechanics: Simulation

- To simulate, define state vector and its derivatives

$$\mathbf{y} = \begin{pmatrix} \mathbf{c} \\ \mathbf{q} \\ \mathbf{v} \\ \omega' \end{pmatrix} \qquad \dot{\mathbf{y}} = \begin{pmatrix} \mathbf{v} \\ \frac{1}{2}\mathbf{q} \circ \begin{pmatrix} 0 \\ \omega' \end{pmatrix} \\ m^{-1}\mathbf{f} \\ \mathbf{I}'^{-1}\left(\tau' - \omega' \times (\mathbf{I}'\omega')\right) \end{pmatrix}$$

- Simulation code available on website

Constrained Dynamics

Sources of Constraints:

- distance and orientation between parts (e.g., joints)
- parameter constraints (e.g., unit norm quaternions)
- limited range of motion (e.g., joint limits)
- interpenetration constraints (e.g., ground contact, self intersection)

Types of Explicit Constraints:

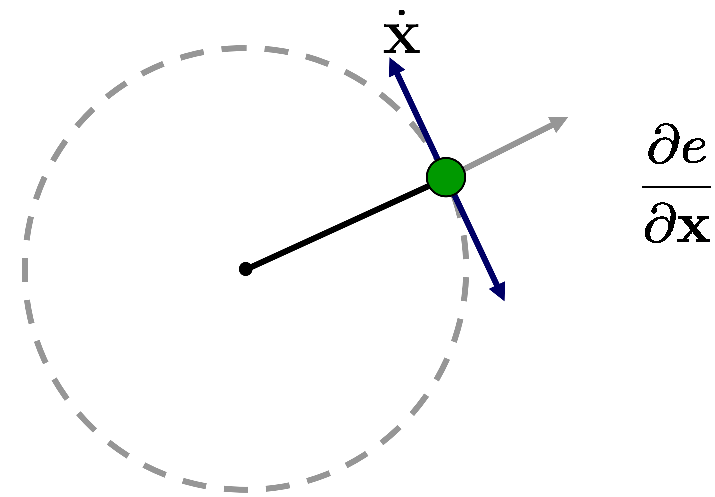
1. $e(\mathbf{x}) = 0$
2. $e(\mathbf{x}) \geq 0$

Constrained Dynamics: Explicit Constraints

Consider a ball on a rod

- State $\mathbf{x}$ is the position of the ball
- Constraint function:

$$e(\mathbf{x}) = \mathbf{x}^T\mathbf{x} - 1 = 0$$



- Admissible state derivatives:

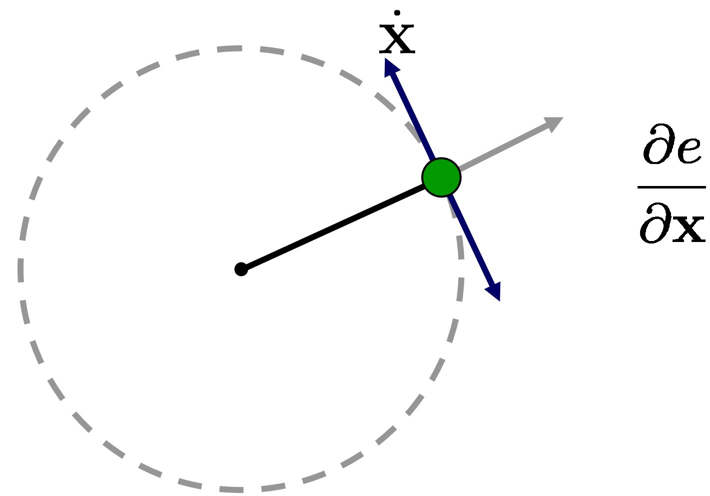
$$\dot{e}(\mathbf{x}) = \frac{\partial e}{\partial \mathbf{x}}\dot{\mathbf{x}} = 0 \qquad \ddot{e}(\mathbf{x}) = \frac{\partial \dot{e}}{\partial \mathbf{x}}\dot{\mathbf{x}} + \frac{\partial e}{\partial \mathbf{x}}\ddot{\mathbf{x}} = 0$$

- Equations of motion:

$$m\ddot{\mathbf{x}} = \mathbf{f} + \mathbf{f}_c$$

external forces

Constrained Dynamics: Explicit Constraints



Consider a ball on a rod

- State $\mathbf{x}$ is the position of the ball
- Constraint function:

$$e(\mathbf{x}) = \mathbf{x}^T\mathbf{x} - 1 = 0$$

- Admissible state derivatives:

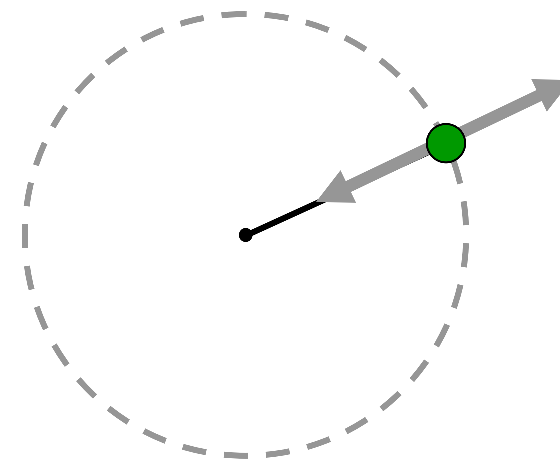
$$\dot{e}(\mathbf{x}) = \frac{\partial e}{\partial \mathbf{x}}\dot{\mathbf{x}} = 0 \qquad \ddot{e}(\mathbf{x}) = \frac{\partial \dot{e}}{\partial \mathbf{x}}\dot{\mathbf{x}} + \frac{\partial e}{\partial \mathbf{x}}\ddot{\mathbf{x}} = 0$$

- Equations of motion:

$$m\ddot{\mathbf{x}} = \mathbf{f} + \mathbf{f}_c$$

constraint forces

Constrained Dynamics: Explicit Constraints



Principle of Virtual Work

- Constraint forces must do no work for every admissible velocity

$$\delta W = \mathbf{f}_c^T \dot{\mathbf{x}} = 0 \qquad \forall \dot{\mathbf{x}},\ \frac{\partial e}{\partial \mathbf{x}} \dot{\mathbf{x}} = 0$$

- Constraint force is proportional to the constraint gradient

$$\mathbf{f}_c = \lambda \frac{\partial e}{\partial \mathbf{x}}$$

- Adding constraint on accelerations gives the augmented equations of motion

$$\begin{pmatrix} m\mathbf{E} & \frac{\partial e}{\partial \mathbf{x}}^T \\ \frac{\partial e}{\partial \mathbf{x}} & 0 \end{pmatrix} \begin{pmatrix} \ddot{\mathbf{x}} \\ \lambda \end{pmatrix} = \begin{pmatrix} \mathbf{f} \\ -\frac{\partial \dot{e}}{\partial \mathbf{x}} \dot{\mathbf{x}} \end{pmatrix}$$

Constrained Dynamics: Explicit Constraints

General Constraints

- For a system with equations of motion

$$\mathbf{M}(\mathbf{x})\ddot{\mathbf{x}} = \mathbf{f}(\mathbf{x},\dot{\mathbf{x}}) + \mathbf{f}_c$$

- with N constraints

$$\mathbf{e}(\mathbf{x}) = \begin{pmatrix} e_1(\mathbf{x}) \\ \vdots \\ e_N(\mathbf{x}) \end{pmatrix} = \mathbf{0}$$

- the augmented equations of motion are

$$\begin{pmatrix} \mathbf{M}(\mathbf{x}) & \frac{\partial \mathbf{e}}{\partial \mathbf{x}}^T \\ \frac{\partial \mathbf{e}}{\partial \mathbf{x}} & 0 \end{pmatrix} \begin{pmatrix} \ddot{\mathbf{x}} \\ \lambda \end{pmatrix} = \begin{pmatrix} \mathbf{f}(\mathbf{x},\dot{\mathbf{x}}) \\ -\frac{\partial \dot{\mathbf{e}}}{\partial \mathbf{x}}\dot{\mathbf{x}} \end{pmatrix}$$

Constrained Dynamics: Explicit Constraints

Quaternion Equations of Motion

- Can use this to write equations of motion directly in terms of quaternions

$$\begin{pmatrix} 4\mathbf{Q}\mathbf{J}\mathbf{Q}^T & 2\mathbf{q} \\ 2\mathbf{q}^T & 0 \end{pmatrix} \begin{pmatrix} \ddot{\mathbf{q}} \\ \lambda \end{pmatrix} = \begin{pmatrix} 2\mathbf{Q}\begin{pmatrix} 0 \\ \tau' \end{pmatrix} + 8\dot{\mathbf{Q}}\mathbf{J}\dot{\mathbf{Q}}^T\mathbf{q} \\ -2\|\mathbf{q}\|^2 \end{pmatrix}$$

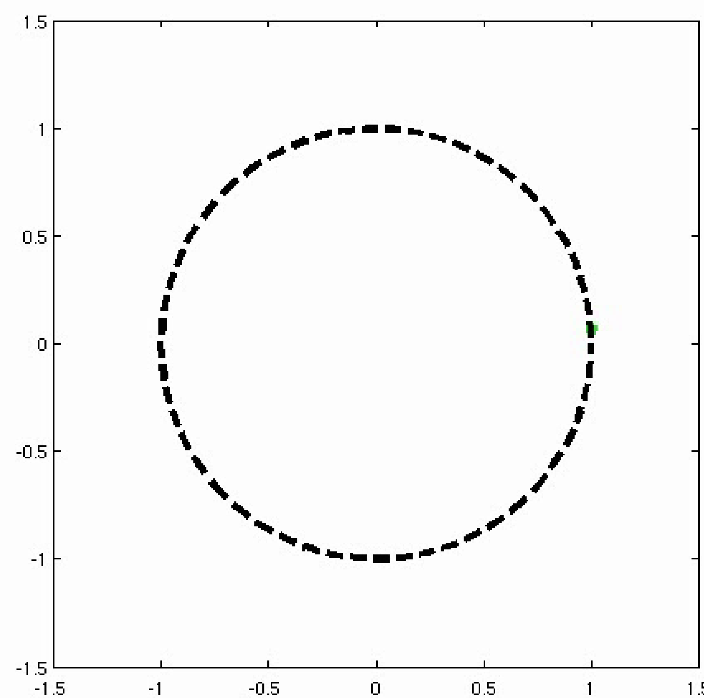
$$\mathbf{J} = \begin{pmatrix} 0 & 0 \\ 0 & \mathbf{I}' \end{pmatrix}$$

(Details are in report.)

Constrained Dynamics: Explicit Constraints

Problems with explicit constraint enforcement

- When integrated numerically, constraints drift
- Dimensionality of state is much larger than effective dimension of constrained space
- Bad for articulation constraints



Constrained Dynamics: Generalized Coordinates

Generalized Coordinates

- A set of coordinates $\mathbf{u}$ which exactly specify the state of a constrained system $\mathbf{x}(\mathbf{u})$
- Admissible velocities and accelerations

$$\begin{aligned} \dot{\mathbf{x}} &= \frac{\partial \mathbf{x}}{\partial \mathbf{u}} \dot{\mathbf{u}} \\ &= \mathbf{T}(\mathbf{u})\dot{\mathbf{u}} \end{aligned} \qquad \ddot{\mathbf{x}} = \mathbf{T}(\mathbf{u})\ddot{\mathbf{u}} + \dot{\mathbf{T}}(\mathbf{u}, \dot{\mathbf{u}})\dot{\mathbf{u}}$$

- Equations of motion

$$\mathbf{M}(\mathbf{x})\ddot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \dot{\mathbf{x}}) + \mathbf{f}_c$$

Constrained Dynamics: Generalized Coordinates

Principle of Virtual Work (again)

- Constraint forces must do no work for every admissible velocity

$$
\begin{aligned}
\delta W &= \mathbf{f}_c^T \dot{\mathbf{x}} = 0 \\
&= \mathbf{f}_c^T \mathbf{T}(\mathbf{u})\dot{\mathbf{u}} = 0 \qquad \forall \dot{\mathbf{u}} \\
&\Rightarrow \mathbf{f}_c^T \mathbf{T}(\mathbf{u}) = 0
\end{aligned}
$$

- Premultiply equations of motion

$$\mathbf{T}(\mathbf{u})^T \mathbf{M}(\mathbf{x})\ddot{\mathbf{x}} = \mathbf{T}(\mathbf{u})^T \mathbf{f}(\mathbf{x}, \dot{\mathbf{x}}) + \cancel{\mathbf{T}(\mathbf{u})^T \mathbf{f}_c}$$

- Substitute in the generalized coordinates and…

Constrained Dynamics: Generalized Coordinates

Equations of Motion for Generalized Coordinates

$$\mathbf{T}(\mathbf{u})^T \mathbf{M}(\mathbf{u}) \mathbf{T}(\mathbf{u}) \ddot{\mathbf{u}} = \mathbf{T}(\mathbf{u})^T \left(\mathbf{f}(\mathbf{u}, \dot{\mathbf{u}}) - \mathbf{M}(\mathbf{u}) \dot{\mathbf{T}}(\mathbf{u}, \dot{\mathbf{u}}) \dot{\mathbf{u}} \right)$$

- Or, more compactly

$$\mathcal{M}(\mathbf{u}) \ddot{\mathbf{u}} = \mathbf{T}(\mathbf{u})^T \mathbf{f}(\mathbf{u}, \dot{\mathbf{u}}) + \mathfrak{g}(\mathbf{u}, \dot{\mathbf{u}})$$

Constrained Dynamics in Generalized Coordinates

Other explicit constraints can still be used with generalized coordinates

$$\begin{pmatrix} \mathcal{M}(\mathbf{u}) & \frac{\partial \mathbf{e}}{\partial \mathbf{u}}^T \\ \frac{\partial \mathbf{e}}{\partial \mathbf{u}} & 0 \end{pmatrix} \begin{pmatrix} \ddot{\mathbf{u}} \\ \lambda \end{pmatrix} = \begin{pmatrix} \mathbf{T}(\mathbf{u})^T \mathbf{f}(\mathbf{u}, \dot{\mathbf{u}}) + \mathfrak{g}(\mathbf{u}, \dot{\mathbf{u}}) \\ -\frac{\partial \dot{\mathbf{e}}}{\partial \mathbf{u}} \dot{\mathbf{u}} \end{pmatrix}$$

This is particularly useful for quaternions and transient attachment constraints

Dynamics of Articulated Rigid Bodies

Equations of Motion for Articulated Rigid Bodies

1. Define parts
2. Define generalized coordinates
3. Define other constraints
4. Differentiate kinematic transformation and constraints

Dynamics of Articulated Rigid Bodies

1. Define Parts

For each part i need to know

- Mass m_i
- Inertia tensor in the body frame I_i'
- Body frame (i.e., center of mass and principal axes)

The biomechanics literature provides this information

Dynamics of Articulated Rigid Bodies

2. Define Generalized Coordinates

Select a root node (e.g., the pelvis)

Define the joints of the body

Generalized coordinates u are: the pose of the root node plus joint angles

Define the kinematic transform $\mathrm{x}(\mathrm{u})$ from the generalized coordinates u to the pose of each body part

Dynamics of Articulated Rigid Bodies

3. Define Constraints

   Quaternion norm constraint for joint angles

   Attachment constraints (e.g., hands or feet)

Dynamics of Articulated Rigid Bodies

4. Differentiate constraints and kinematic transformation

$$\mathbf{T}(\mathbf{u}) = \frac{\partial \mathbf{x}}{\partial \mathbf{u}} \qquad \frac{\partial \mathrm{e}}{\partial \mathbf{u}}$$

$$\dot{\mathbf{T}}(\mathbf{u}, \dot{\mathbf{u}}) = \sum_i \frac{\partial^2 \mathbf{x}}{\partial \mathbf{u} \partial u_i} \dot{u}_i \qquad \frac{\partial \dot{\mathrm{e}}}{\partial \mathbf{u}} = \sum_i \frac{\partial^2 \mathrm{e}}{\partial \mathbf{u} \partial u_i} \dot{u}_i$$

Good News:

We`re giving Matlab code to do all of this for you.