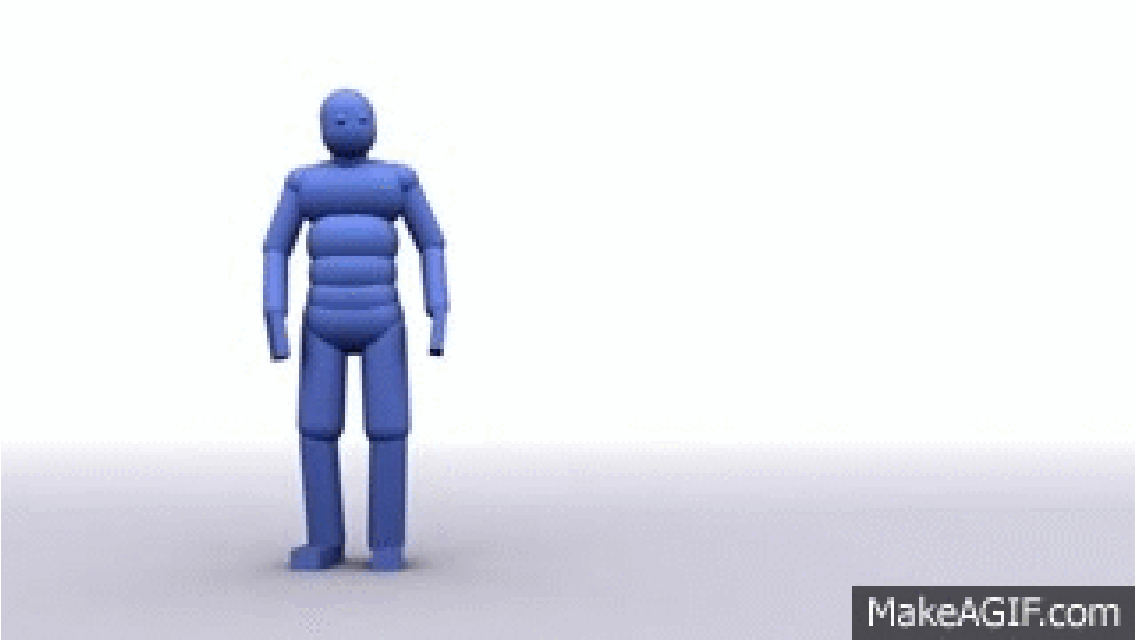
ETH zürich



Articulated Rigid Body Systems

Constrained dynamics and generalized coordinates formulations

CRL

Learning objectives

- Learn how to model articulated rigid body dynamics using maximal coordinates (i.e. explicit and implicit penalty forces, velocity-level constraints), as well as reduced formulations

Rigid Body Dynamics

- At each time step:
  - Compute net force $\boldsymbol{F}$ and net torque $\boldsymbol{\tau}$ acting on the rigid body
  - Update linear and angular velocities:

$$\boldsymbol{v}_{\mathrm{i+1}} = \boldsymbol{v}_{\mathrm{i}} + h\frac{\boldsymbol{F}}{M}$$

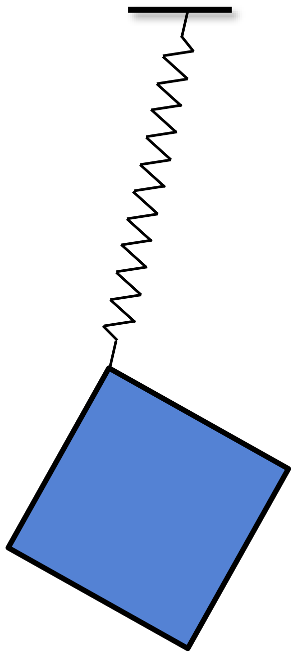
$$\boldsymbol{\omega}_{\mathrm{i+1}} = \boldsymbol{\omega}_{\mathrm{i}} + h\boldsymbol{I}^{-1}(\boldsymbol{\tau} - \boldsymbol{\omega_i} \times \boldsymbol{I}\boldsymbol{\omega_i})$$

  - Update COM position:

$$\boldsymbol{p}_{\mathrm{i+1}} = \boldsymbol{p}_{\mathrm{i}} + h\boldsymbol{v}_{\mathrm{i+1}}$$

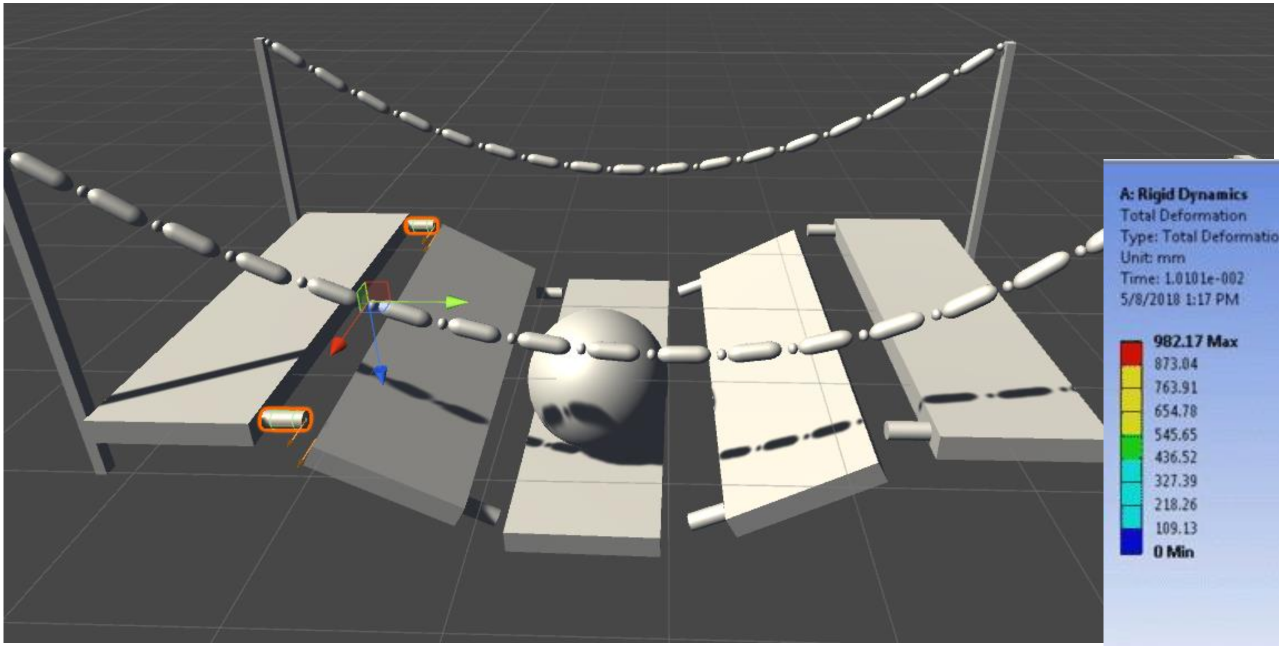
  - Update rigid body orientation:

$$\boldsymbol{q}_{\mathrm{i+1}} = \mathbf{q}\left(h|\boldsymbol{\omega}_{\mathrm{i+1}}|, \frac{\boldsymbol{\omega}_{\mathrm{i+1}}}{|\boldsymbol{\omega}_{\mathrm{i+1}}|}\right)\boldsymbol{q}_{\mathrm{i}}$$



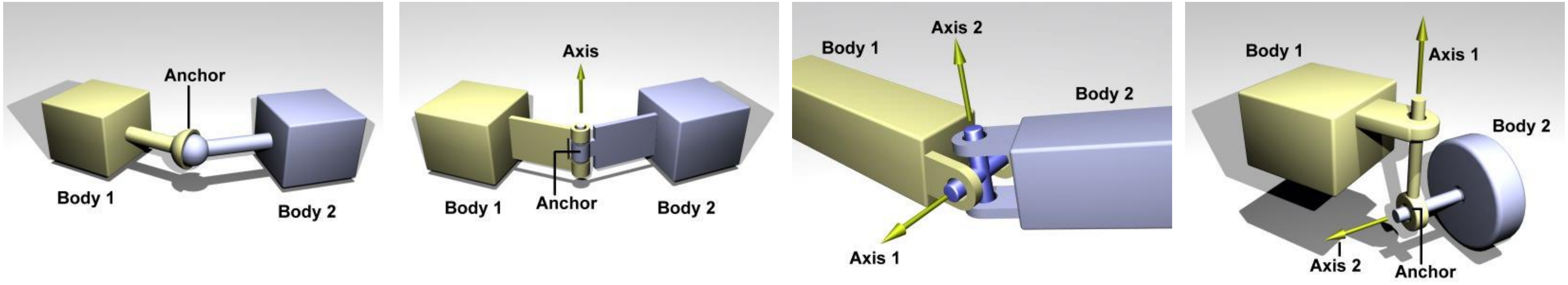
Multi-body systems

- We often want to model the physics of *articulated rigid body systems*



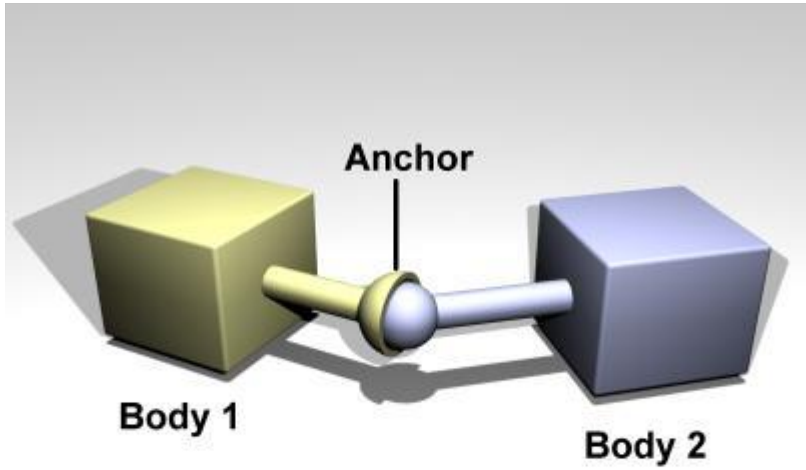
Multi-body systems

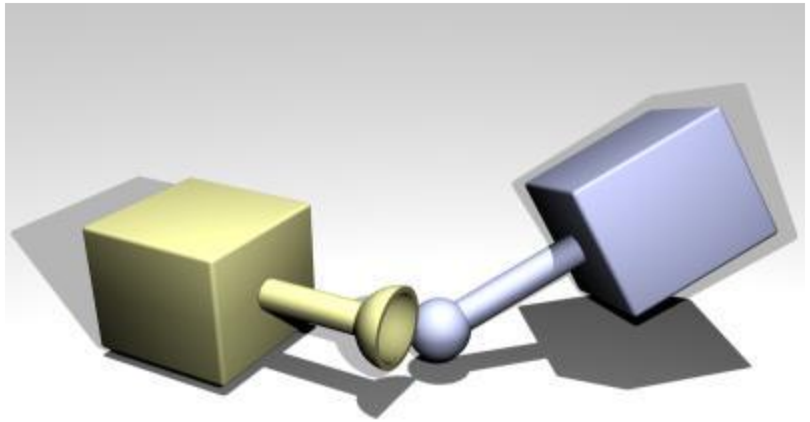
- We often want to model the physics of *articulated rigid body systems*
  - Collections of rigid bodies that are interconnected through joints. The joints anchor pairs of rigid bodies to each other – they restrict the way they can move relative to each other



Multi-body systems

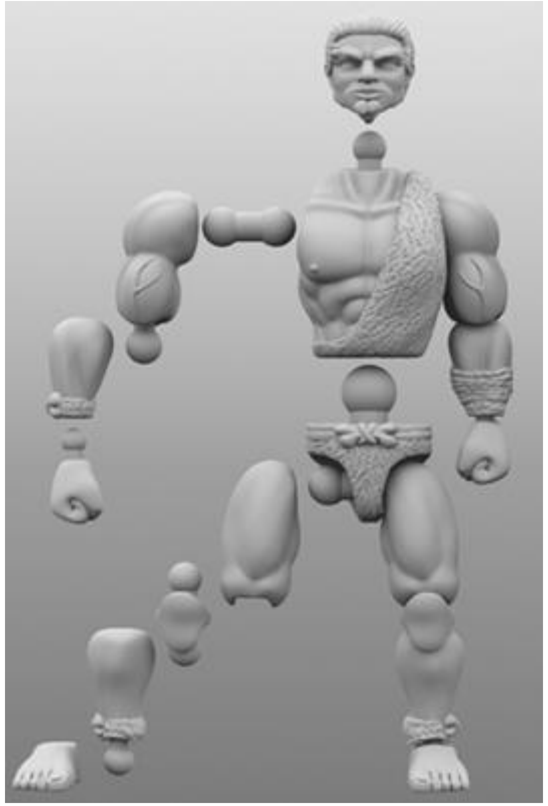
- We often want to model the physics of *articulated rigid body systems*
  - Collections of rigid bodies that are interconnected through joints. The joints anchor pairs of rigid bodies to each other – they restrict the way they can move relative to each other.
  - We can therefore talk about valid and invalid configurations of a multi-body system
- A reasonable simulation engine should produce motions for the multi-body system that respect, to the extent possible, all articulation constraints

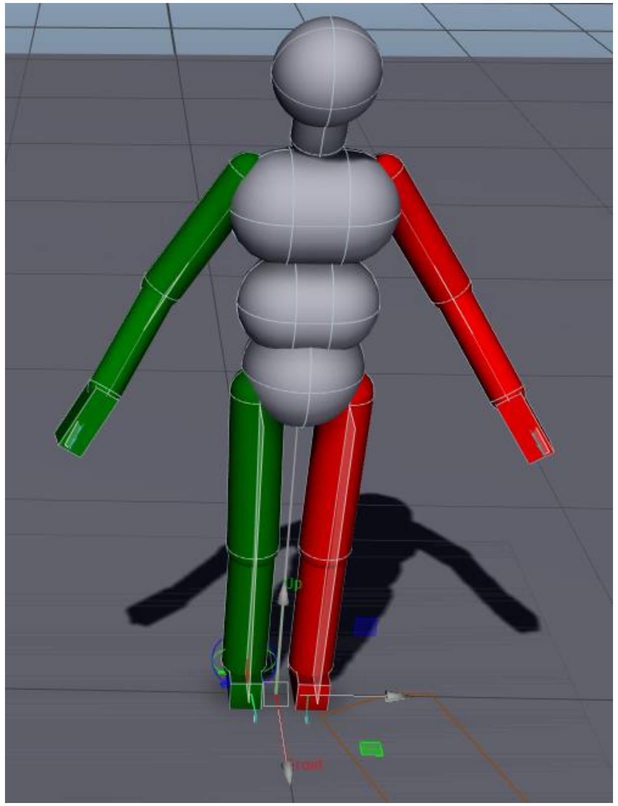




Multi-body systems

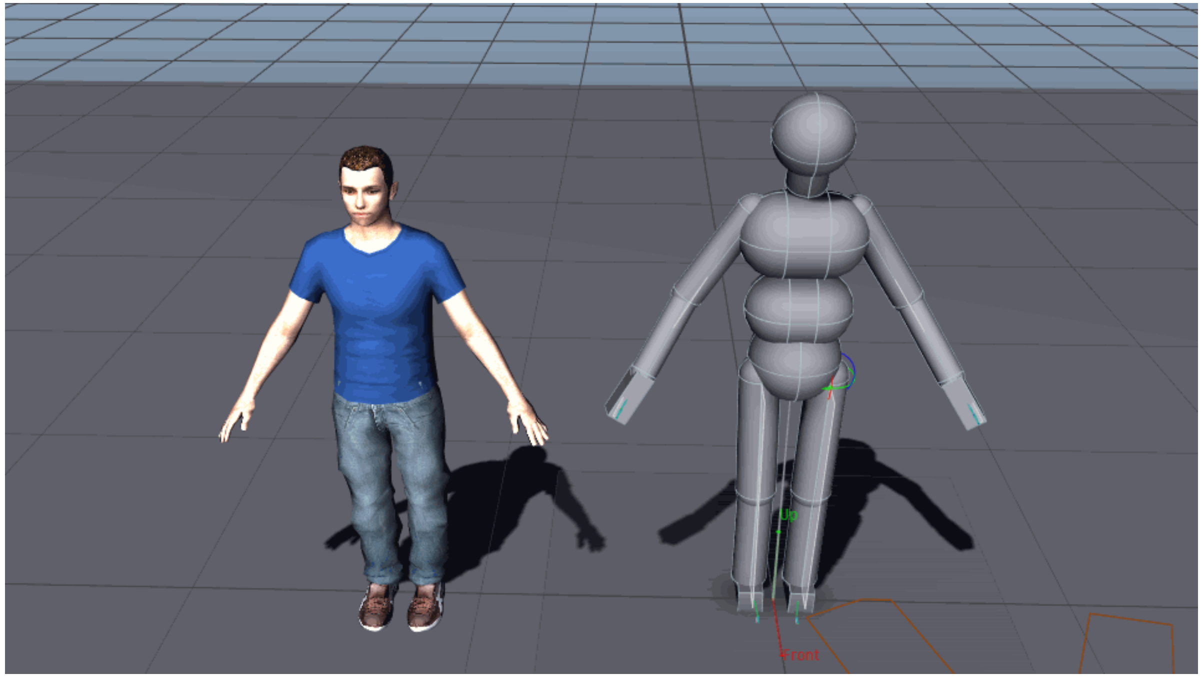
- We often want to model the physics of *articulated rigid body systems*

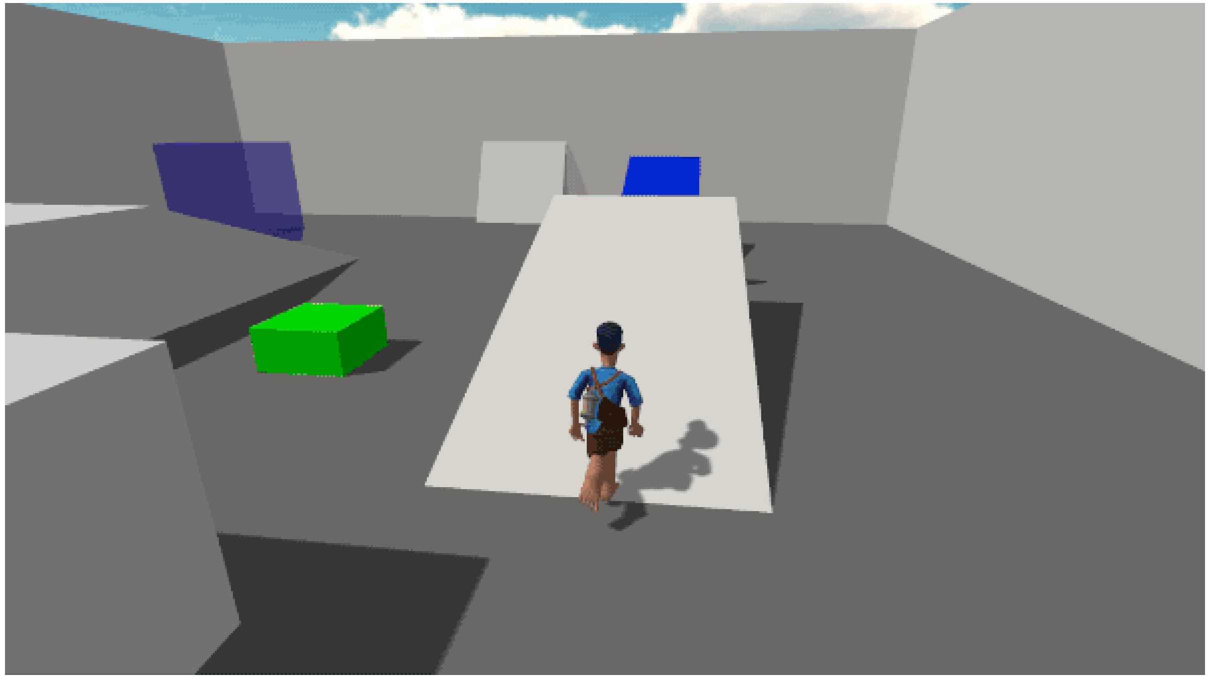




Multi-body systems

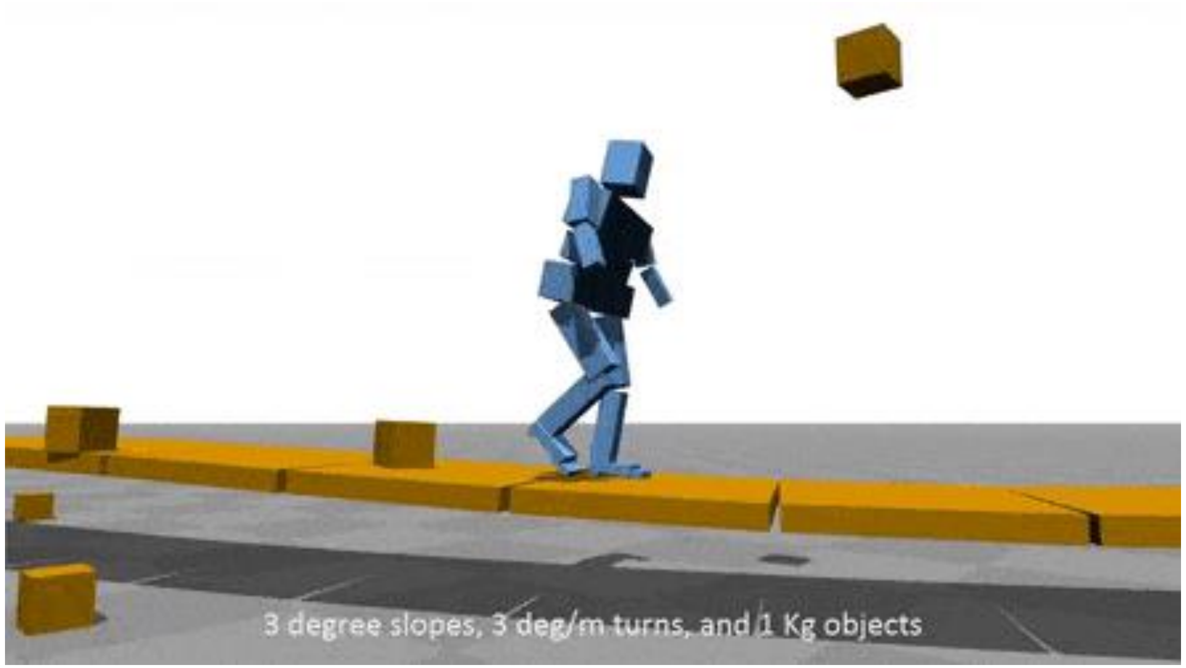
- We often want to model the physics of *articulated rigid body systems*

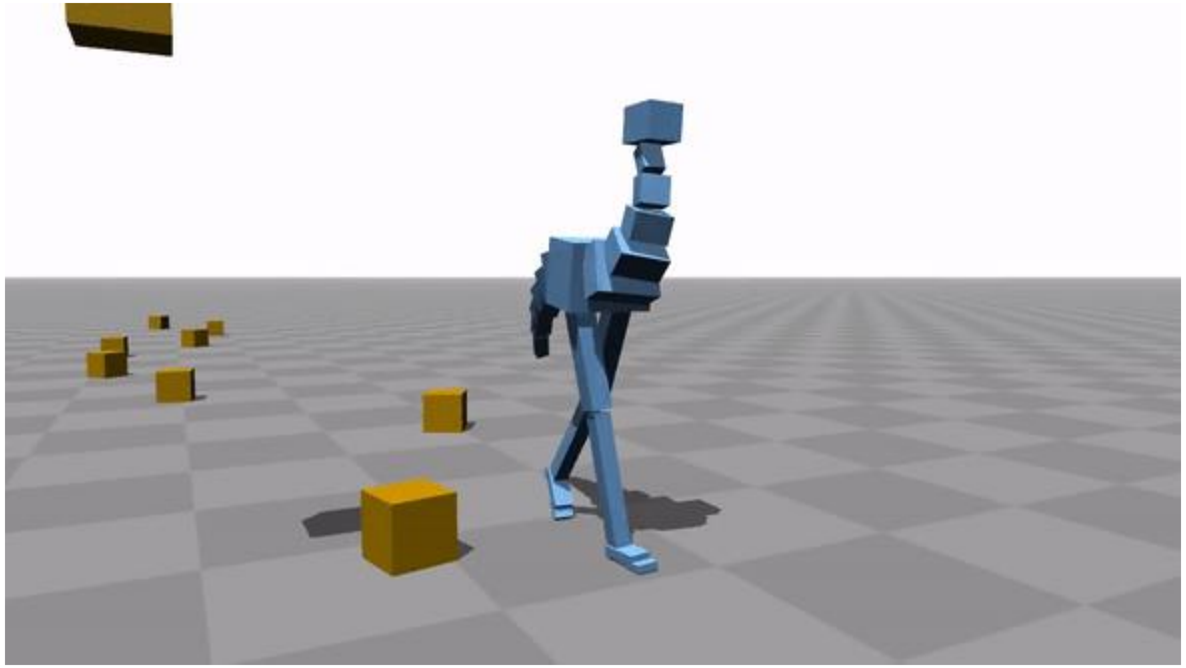




Multi-body systems

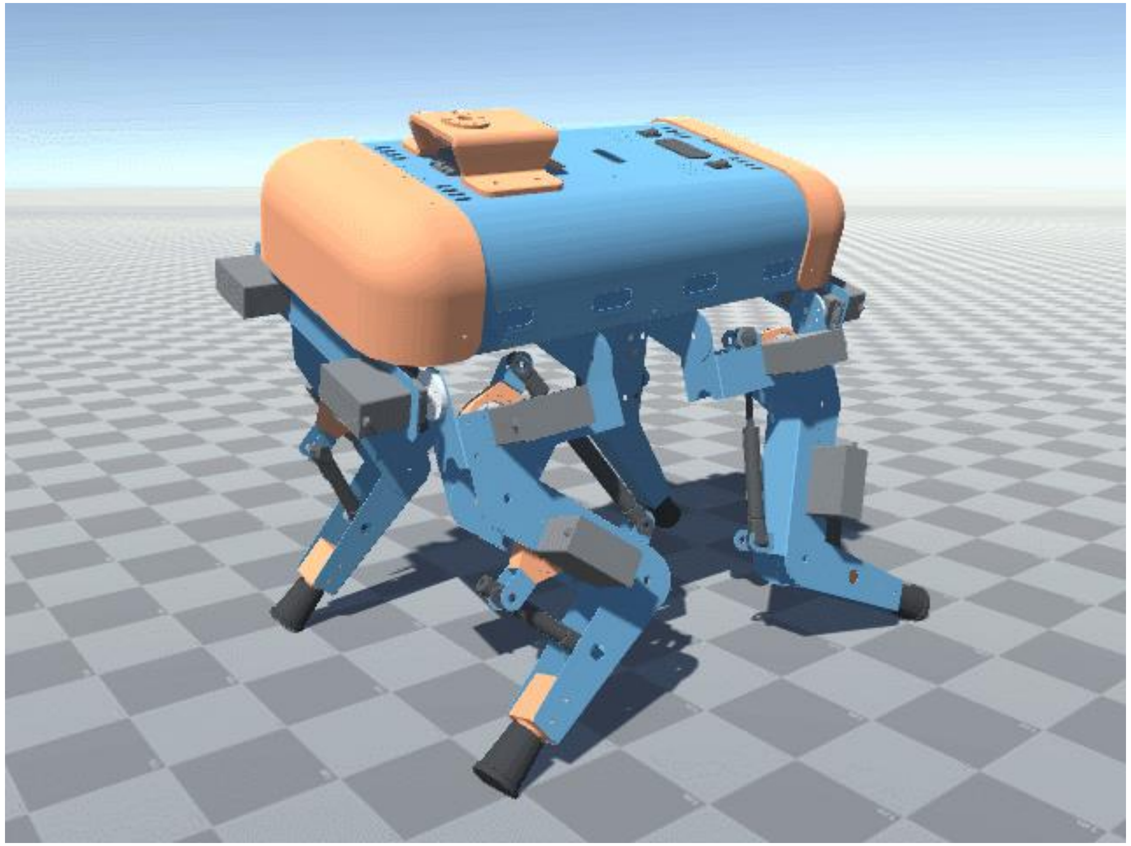
- We often want to model the physics of *articulated* rigid body systems

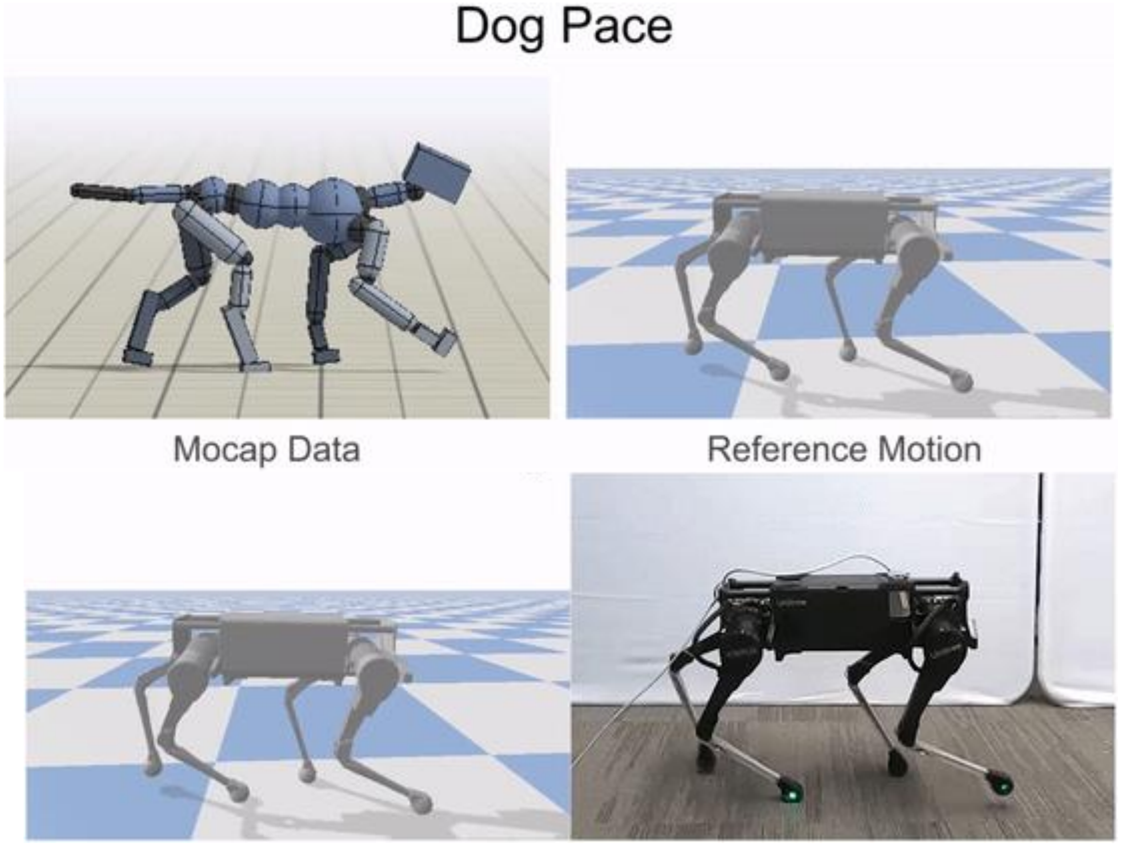




Multi-body systems

- We often want to model the physics of *articulated* rigid body systems



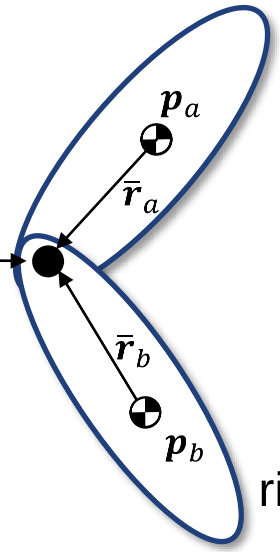


Multi-body systems

- So, how do we go about modeling multi-body systems?

Global coordinates of the joint:

$\boldsymbol{x}_a = \boldsymbol{p}_a + \boldsymbol{R}_a\bar{\boldsymbol{r}}_a$ and/or $\boldsymbol{x}_\mathrm{b} = \boldsymbol{p}_b + \boldsymbol{R}_b\bar{\boldsymbol{r}}_b$

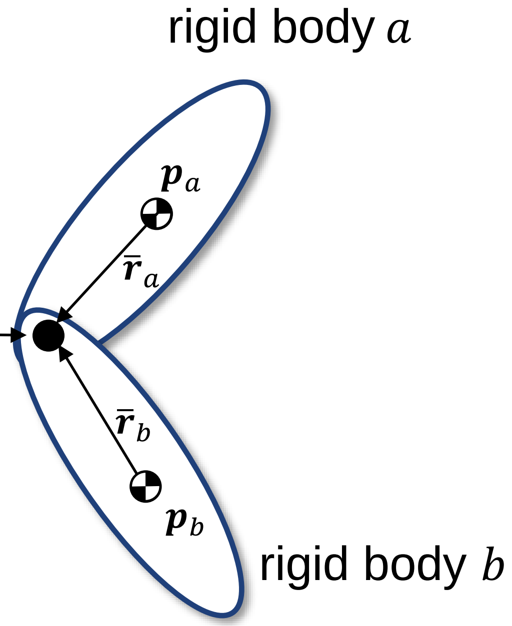


Multi-body systems

- So, how do we go about modeling multi-body systems?

Global coordinates of the joint:

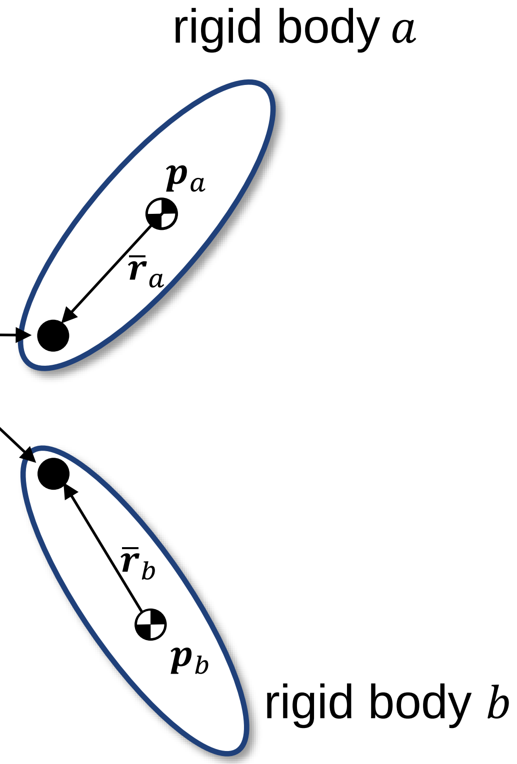
$\boldsymbol{x}_a = \boldsymbol{p}_a + \boldsymbol{R}_a\bar{\boldsymbol{r}}_a$ and/or $\boldsymbol{x}_\mathrm{b} = \boldsymbol{p}_b + \boldsymbol{R}_b\bar{\boldsymbol{r}}_b$



Multi-body systems

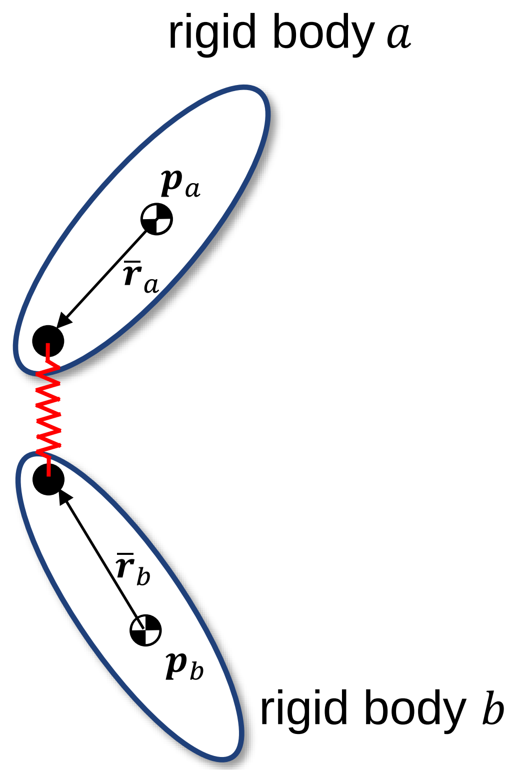
- So, how do we go about modeling multi-body systems?

Global coordinates of the joint:
$\boldsymbol{x}_a = \boldsymbol{p}_a + \boldsymbol{R}_a \bar{\boldsymbol{r}}_a$ and/or $\boldsymbol{x}_\mathrm{b} = \boldsymbol{p}_b + \boldsymbol{R}_b \bar{\boldsymbol{r}}_b$



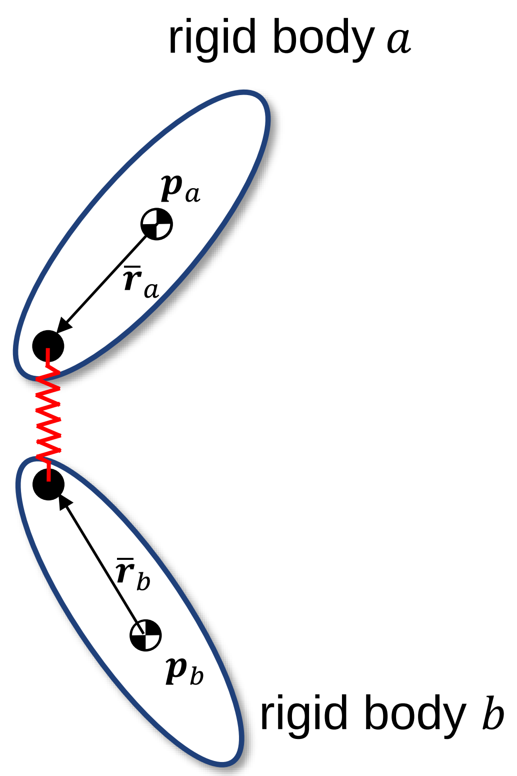
Modeling multi-body systems – a first attempt

- So, how do we go about modeling multi-body systems?
  - Imagine there is a (zero rest length) rubber band / spring connecting the two pins of the joint
  - Compute the force generated by the tension in the spring, apply to the two rigid bodies (equal and opposite!), and integrate forward in time



Modeling multi-body systems – a first attempt

- So, how do we go about modeling multi-body systems?
  - Imagine there is a (zero rest length) rubber band / spring connecting the two pins of the joint
  - Compute the force generated by the tension in the spring, apply to the two rigid bodies (equal and opposite!), and integrate forward in time
  - Not too bad of an approximation – e.g. ligaments that connect our bones to each other are essentially (very stiff) springs
  - High spring stiffness (treated as explicit penalty terms) causes numerical stability problems
    - Must carefully trade off size of time step vs gain stiffness/drift

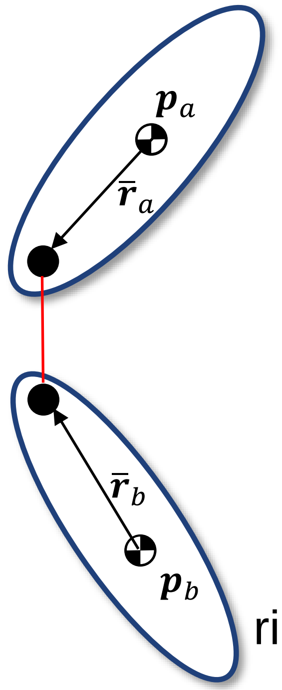


Modeling multi-body systems – attempt #2

- A very general velocity-level constraint-based formulation
  - Step 1: define a vector-valued function $\mathbf{C}(\mathscr{p})$ that is 0 when the multi-body system is in a valid configuration; $\mathscr{p}$ denotes the position and orientation of all rigid bodies stacked together.

Modeling multi-body systems – attempt #2

- $\mathbf{C}(\wp)$:

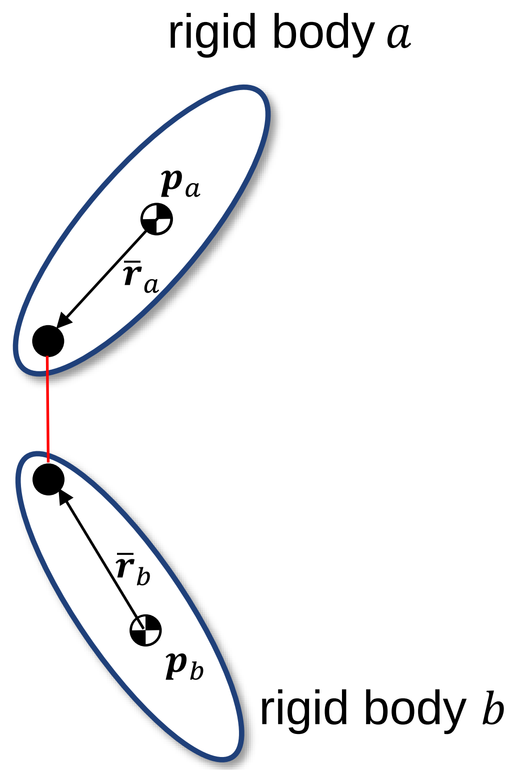


Modeling multi-body systems – attempt #2

- A very general velocity-level constraint-based formulation
  - Step 1: define a vector-valued function $\mathbf{C}(\mathscr{p})$ that is 0 when the multi-body system is in a valid configuration; $\mathscr{p}$ denotes the position and orientation of all rigid bodies stacked together.
  - Step 2: note that $\dot{\mathbf{C}} = \frac{d\mathbf{C}}{dt} = A\mathscr{v}$; A is the Jacobian of the constraint vector $\mathbf{C}$, and $\mathscr{v}$ is a vector that holds linear and angular velocities for all rigid bodies in the system

Modeling multi-body systems – attempt #2

- $\dot{\mathrm{C}} = A\boldsymbol{v}$:

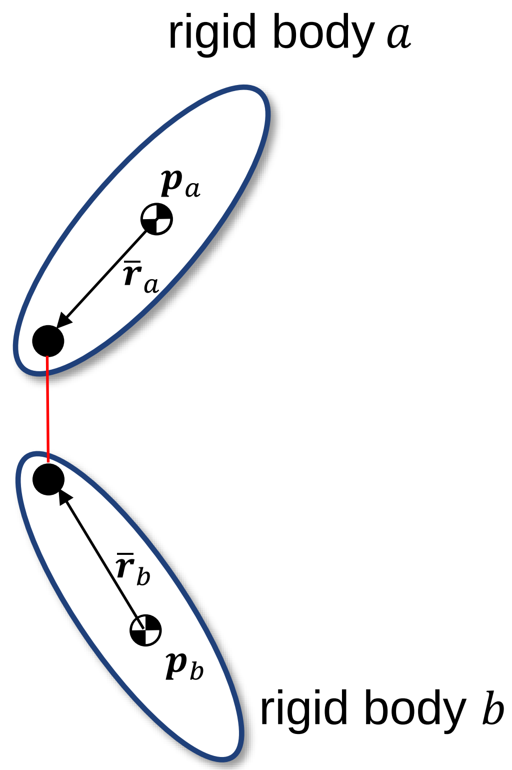


Modeling multi-body systems – attempt #2

- A very general velocity-level constraint-based formulation
  - Step 1: define a vector-valued function $\mathbf{C}(\mathcal{p})$ that is 0 when the multi-body system is in a valid configuration; $\mathcal{p}$ denotes the position and orientation of all rigid bodies stacked together.
  - Step 2: note that $\dot{\mathbf{C}} = \frac{d\mathbf{C}}{dt} = A\mathcal{v}$; A is the Jacobian of the constraint vector $\mathbf{C}$, and $\mathcal{v}$ is a vector that holds linear and angular velocities for all rigid bodies in the system
  - Step 3: note update rule for system velocities in vector form: $\mathcal{v}_{t+1} = \mathcal{v}_t + hM^{-1}F$; F stacks all forces and torques in a vector, M^{-1} stacks all masses and moment of inertia tensors in a matrix

Modeling multi-body systems – attempt #2

- $v_{t+1} = v_t + hM^{-1}F$

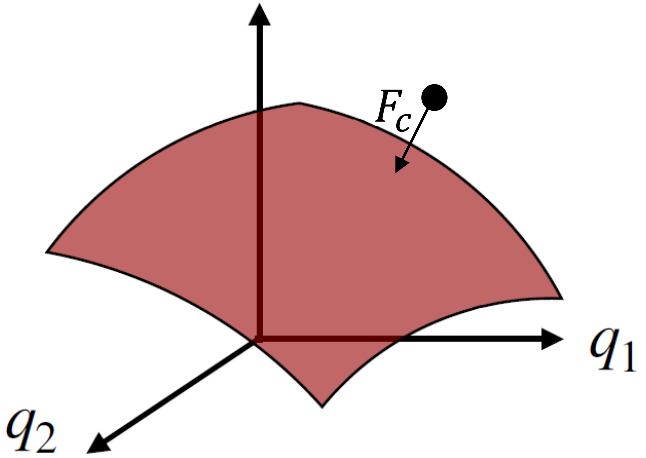


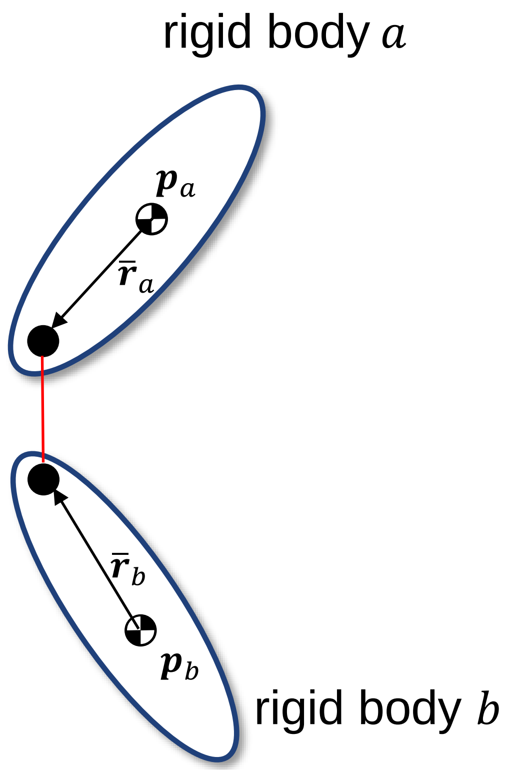
Modeling multi-body systems – attempt #2

- A very general velocity-level constraint-based formulation
  - Step 1: define a vector-valued function $\mathbf{C}(\mathscr{p})$ that is 0 when the multi-body system is in a valid configuration; $\mathscr{p}$ denotes the position and orientation of all rigid bodies stacked together.
  - Step 2: note that $\dot{\mathbf{C}} = \frac{d\mathbf{C}}{dt} = A\mathscr{v}$; A is the Jacobian of the constraint vector $\mathbf{C}$, and $\mathscr{v}$ is a vector that holds linear and angular velocities for all rigid bodies in the system
  - Step 3: note update rule for system velocities in vector form: $\mathscr{v}_{t+1} = \mathscr{v}_t + hM^{-1}F$; F stacks all forces and torques in a vector, M^{-1} stacks all masses and moment of inertia tensors in a matrix
  - Step 4: note structure of F: $F = F_{ext} + F_c$, where the constraint forces are defined as $F_c = A^t\lambda$ according to the principle of virtual work

Modeling multi-body systems – attempt #2

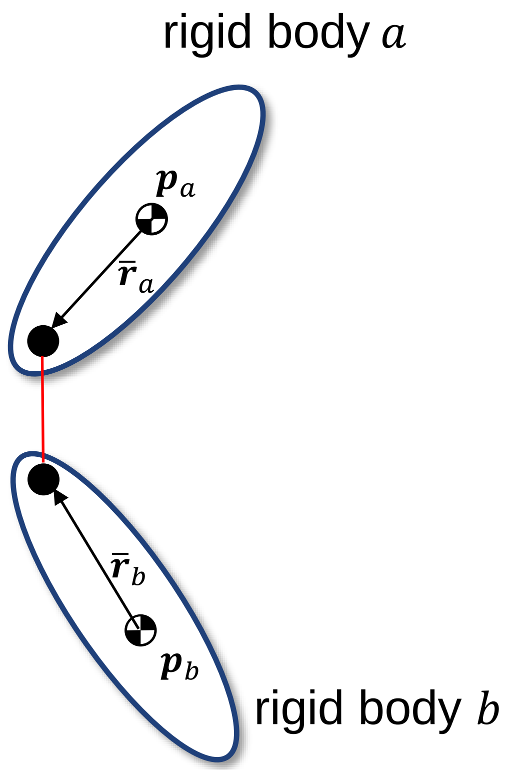
- Principle of virtual work: $F_c = A^t \lambda$
  - Constraint forces must "pull" the system towards the constraint manifold in the most direct way possible





Modeling multi-body systems – attempt #2

- Principle of virtual work: $F_c = A^t \lambda$
  - Can also think of this as a way of mapping forces from the space defined by the constraints into the space defined by the configuration of the entire multi-body system

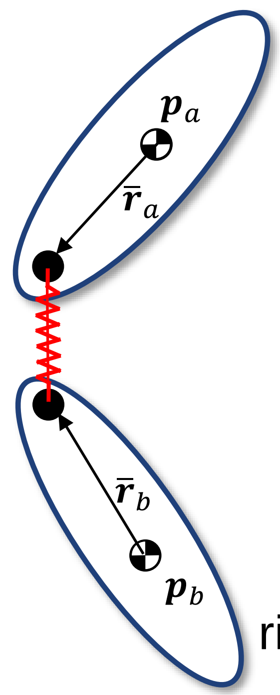


Modeling multi-body systems – attempt #2

- A very general velocity-level constraint-based formulation
  - Step 1: define a vector-valued function $\mathbf{C}(\mathscr{p})$ that is 0 when the multi-body system is in a valid configuration; $\mathscr{p}$ denotes the position and orientation of all rigid bodies stacked together.
  - Step 2: note that $\dot{\mathbf{C}} = \frac{d\mathbf{C}}{dt} = A\mathscr{v}$; A is the Jacobian of the constraint vector $\mathbf{C}$, and $\mathscr{v}$ is a vector that holds linear and angular velocities for all rigid bodies in the system
  - Step 3: note update rule for system velocities in vector form: $\mathscr{v}_{t+1} = \mathscr{v}_t + hM^{-1}F$; F stacks all forces and torques in a vector, M^{-1} stacks all masses and moment of inertia tensors in a matrix
  - Step 4: note structure of F: $F = F_{ext} + F_c$, where the constraint forces are defined as $F_c = A^t\lambda$ according to the principle of virtual work
  - Step 5: compute λ, either directly or as a function of a target value for $\dot{\mathbf{C}}_{t+1}$ (aka a velocity-level constraint). Note: $\mathbf{C}_{\mathrm{t+1}} \approx \boldsymbol{C}_t + h\dot{\mathbf{C}}_{t+1}$, $\dot{\mathbf{C}}_t = A\mathscr{v}_t$, $\dot{\mathbf{C}}_{t+1} = A\mathscr{v}_{t+1}$

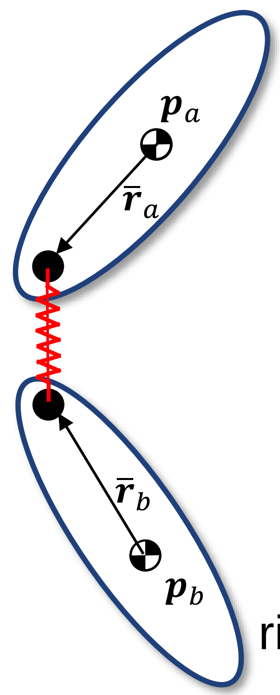
Modeling multi-body systems – attempt #2

- Computing λ



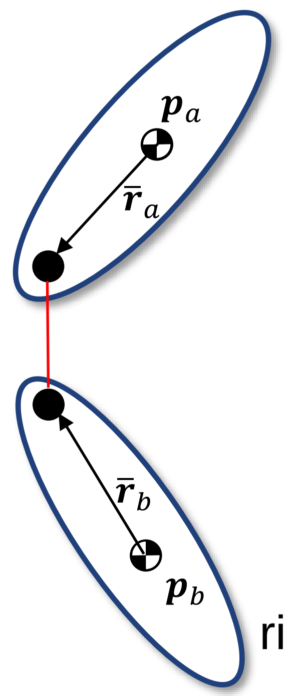
Modeling multi-body systems – attempt #2

- Computing λ



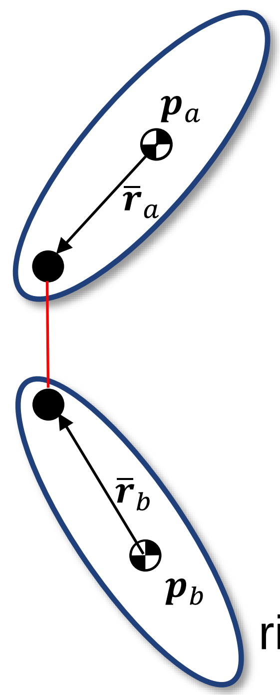
Modeling multi-body systems – attempt #2

- Velocity-level constraints



Modeling multi-body systems – attempt #2

- From target $\dot{\mathbf{C}}_{t+1}$ to λ



Modeling multi-body systems – attempt #2

- A very general velocity-level constraint-based formulation
  - Step 1: define a vector-valued function $\mathbf{C}(\mathscr{p})$ that is 0 when the multi-body system is in a valid configuration; $\mathscr{p}$ denotes the position and orientation of all rigid bodies stacked together.
  - Step 2: note that $\dot{\mathbf{C}} = \frac{d\mathbf{C}}{dt} = A\boldsymbol{v}$; A is the Jacobian of the constraint vector $\mathbf{C}$, and $\boldsymbol{v}$ is a vector that holds linear and angular velocities for all rigid bodies in the system
  - Step 3: note update rule for system velocities in vector form: $\boldsymbol{v}_{t+1} = \boldsymbol{v}_t + hM^{-1}F$; F stacks all forces and torques in a vector, M^{-1} stacks all masses and moment of inertia tensors in a matrix
  - Step 4: note structure of F: $F = F_{ext} + F_c$, where the constraint forces are defined as $F_c = A^t\lambda$ according to the principle of virtual work
  - Step 5: compute λ, either directly or as a function of a target value for $\dot{\mathbf{C}}_{t+1}$ (aka a velocity-level constraint). Note: $\mathbf{C}_{t+1} \approx \boldsymbol{C}_t + h\dot{\mathbf{C}}_{t+1}$, $\dot{\mathbf{C}}_t = A\boldsymbol{v}_t$, $\dot{\mathbf{C}}_{t+1} = A\boldsymbol{v}_{t+1}$
  - Step 6: compute $\boldsymbol{v}_{t+1}$ using update rule, then integrate forward to get new positions and orientations for each rigid body in the system

Modeling multi-body systems – attempt #2

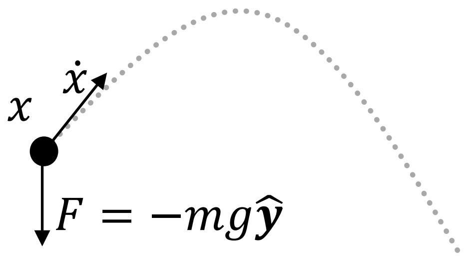
- A very general velocity-level constraint-based formulation
- Easy to implement many types of constraints
  - See for instance https://danielchappuis.ch/download/ConstraintsDerivationRigidBody3D.pdf or https://www10.cs.fau.de/publications/theses/2009/Pickl_MT_2009.pdf
- Implemented in many popular physics engines
  - ODE, Bullet, Gazebo, etc
- Maximal coordinates formulation
  - Explicitly compute and apply constraint forces, use typical EoM for each individual rigid body
  - Very modular, easy to create or break constraints at run-time, etc.
  - Easy to handle kinematic loops, holonomic and non-holonomic constraints, etc
- Formulations based on generalized (or reduced, or minimal) coordinates exist, too
  - Main idea: bake constraints directly into the equations of motion

Lagrangian Mechanics

- Beautifully simple and general recipe:
  - Choose a set of *generalized coordinates* q that describe the system we want to model
    - they should be independent and completely determine the configuration of the system
    - there may be many choices for generalized coordinates for a physical system, and they can have an impact in terms of how convenient the solution of the underlying ODE will be
    - We use the generalized coordinates to define the cartesian coordinates of any point in the system we are modeling, i.e. the map $\mathrm{x}(q)$ must be explicitly specified. The derivatives of the map gives us velocities $\dot{x} = \frac{d\mathrm{x}}{dt} = \frac{\partial \mathrm{x}}{\partial q}\dot{q} = J\dot{q}$
  - Write down the system's kinetic and potential energies, K and U
  - Write down the *Lagrangian* $L \coloneqq K - U$
  - Dynamics then given by *Euler-Lagrange equation* $\frac{d}{dt}\frac{\partial L}{\partial \dot{q}} = \frac{\partial L}{\partial q}$

Lagrangian Mechanics – examples

- Consider a particle moving under gravity



- Choose a set of generalized coordinates: $q \coloneqq x$
- Kinetic energy: $K = \frac{1}{2}\dot{x}^T m \dot{x}$
- Potential energy: $U = mgh = mg\hat{\mathbf{y}}^T x = -F^T x$
- Lagrangian: $L \coloneqq K - U$
- Equations of motion: $\frac{d}{dt}\frac{\partial L}{\partial \dot{q}} = \frac{\partial L}{\partial q}$

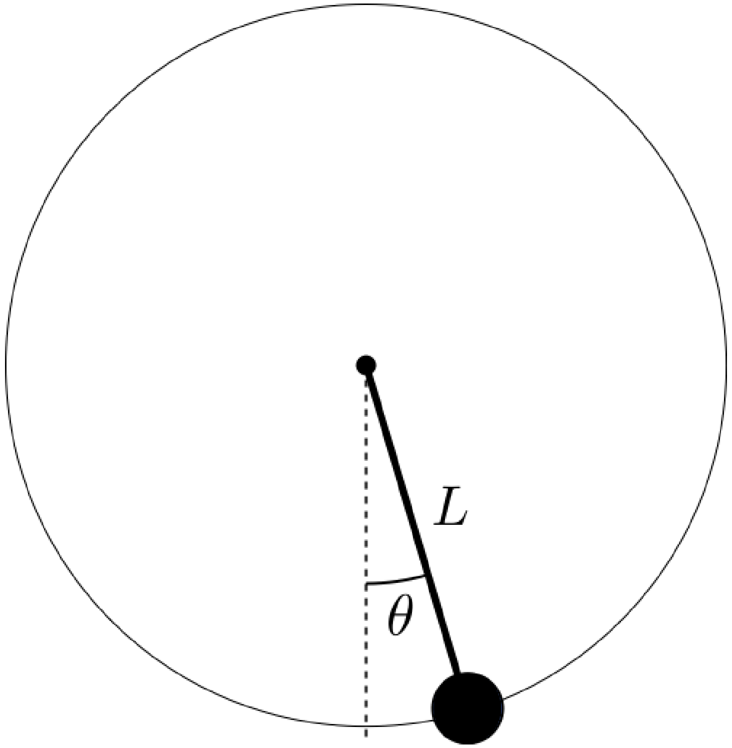
$$\frac{\partial L}{\partial \dot{q}} = \frac{\partial L}{\partial \dot{x}} = m\dot{x} \Rightarrow \frac{d}{dt}\frac{\partial L}{\partial \dot{q}} = m\ddot{x}$$

$$\frac{\partial L}{\partial q} = \frac{\partial L}{\partial x} = -\frac{dU}{dx} = F$$

$$F = m\ddot{x}$$

Lagrangian Mechanics – examples

- Same particle, but it is now constrained to move along a circle (think pendulum, or bead on a wire)



- Choose a set of generalized coordinates: $q \coloneqq \theta$

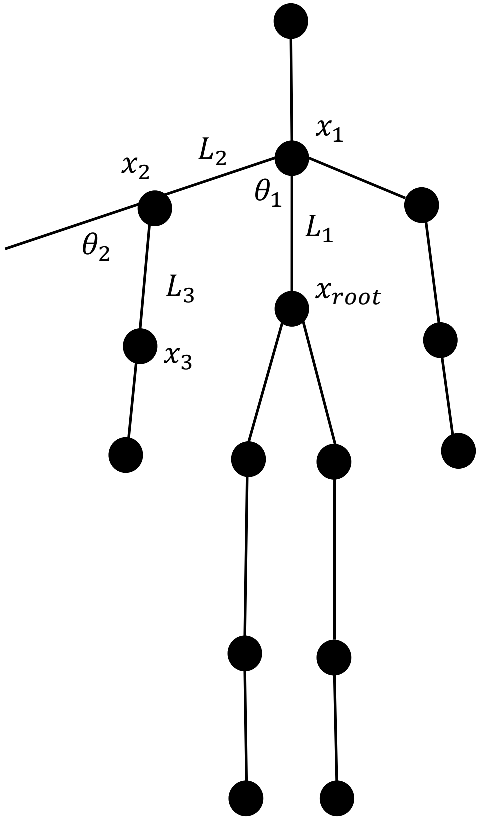
$$x(q) = \begin{bmatrix} L\sin q \\ -L\cos q \end{bmatrix}; \dot{x}(q) = \begin{bmatrix} L\cos q \\ L\sin q \end{bmatrix}\dot{q}$$

- Kinetic energy: $K = \frac{1}{2}\dot{x}^T m\dot{x} = \frac{mL^2}{2}\dot{q}^2$
- Potential energy: $U = mgh = mg\hat{\mathbf{y}}^T x = -mgL\cos q$
- Lagrangian: $L \coloneqq K - U$
- Equations of motion: $\frac{d}{dt}\frac{\partial L}{\partial \dot{q}} = \frac{\partial L}{\partial q} \Rightarrow \ddot{q} = -\frac{g}{L}\sin q$

Lagrangian Mechanics – examples

Now let's consider *many* particles connected to each other!

- Generalized coordinates: $q \coloneqq [x_{root}\ \theta_1\ \theta_2\ \dots]$
- You should be able to compute $\boldsymbol{x}(q) = [x_{root}\ \ x_1\ \ x_2\ \dots]$, $\dot{\boldsymbol{x}} = \frac{\partial \boldsymbol{x}}{\partial q}\dot{q} = J\dot{q}$ using forward kinematics
- Kinetic energy: $K = \frac{1}{2}\dot{\boldsymbol{x}}^T M \dot{\boldsymbol{x}} = \frac{1}{2}\dot{q}^T J^T M J \dot{q}$
- Potential energy: $U = -\boldsymbol{F}^T \boldsymbol{x}$
- Lagrangian: $L \coloneqq K - U$
- Equations of motion: $\frac{d}{dt}\frac{\partial L}{\partial \dot{q}} = \frac{\partial L}{\partial q}$



Lagrangian Mechanics – examples

Let's work out all the necessary derivatives (http://www.matrixcalculus.org/):

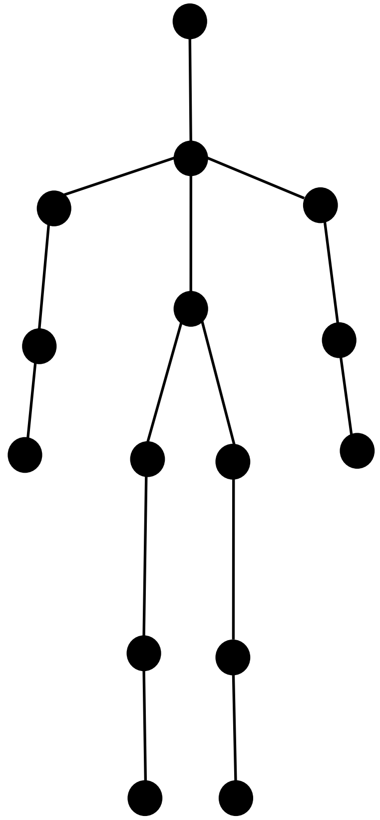
$$K = \frac{1}{2}\dot{q}^T J^T M J \dot{q};\ U = -\boldsymbol{F}^T \boldsymbol{x};\ L = K - U$$

$$\frac{\partial L}{\partial \dot{q}} = \frac{\partial K}{\partial \dot{q}} = J^T M J \dot{q}$$

$$\frac{d}{dt}\frac{\partial L}{\partial \dot{q}} = \dot{J}^T M J \dot{q} + \cancel{J^T \dot{M} J \dot{q}} + J^T M \dot{J} \dot{q} + J^T M J \ddot{q}$$

$$\frac{\partial L}{\partial q} = \frac{\partial K}{\partial q} - \frac{\partial U}{\partial q} = \left(\frac{\partial J}{\partial q}\dot{q}\right)^T M J \dot{q} + J^T \boldsymbol{F} = \dot{J}^T M J \dot{q} + J^T \boldsymbol{F}$$

$$\frac{d}{dt}\frac{\partial L}{\partial \dot{q}} = \frac{\partial L}{\partial q} \quad \Rightarrow \quad \underbrace{J^T M J}_{\text{Generalized mass matrix } \boldsymbol{M}(q)} \ddot{q} + \underbrace{J^T M \dot{J} \dot{q}}_{\text{Inertial effects (e.g. Coriolis and centrifugal forces)}} = \underbrace{J^T \boldsymbol{F}}_{\text{Generalized forces (cartesian-space forces projected into generalized coordinates)}}$$



Lagrangian Mechanics – examples

The same approach can be used to derive the well-known equations of motion for articulated rigid body systems:

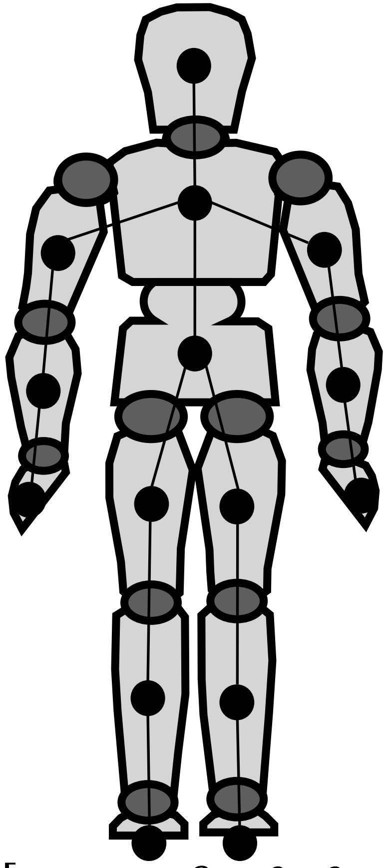
Generalized accelerations

Generalized forces

$$M(q)\ddot{q} + C(q, \dot{q}) = Q$$

Generalized mass matrix

Coriolis and centrifugal terms



$q := [x_{root}\ \alpha\ \ \beta\ \gamma\ \theta_1\ \theta_2 \ldots]$

See “A Quick Tutorial on Multibody Dynamics” by C. Karen Liu and Sumit Jain for a full derivation.

Lagrangian Mechanics

- Elegant formulation based on generalized, or reduced coordinates
  - Allows us to completely eliminate certain types of constraints
- You should work out the dynamics of a rigid body using Lagrangian mechanics
  - A system of particles whose generalized coordinates are the position of the center of mass and degrees of freedom for the orientation (e.g. Euler angles)
  - Concepts we’ve taken for granted, like “torques” (i.e. cartesian forces projected into the generalized coordinates of the ensemble of particles) become much more clear
- We will see the equations of motion for articulated rigid bodies again later on in the course!