Kinematics

CS 448D: Character Animation
Prof. Vladlen Koltun
Stanford University

Kinematics

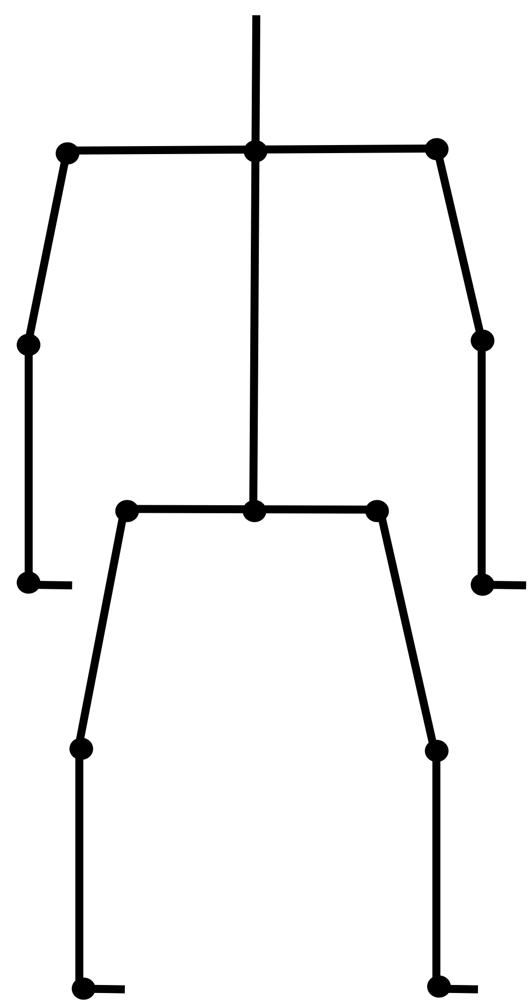
Kinematics:

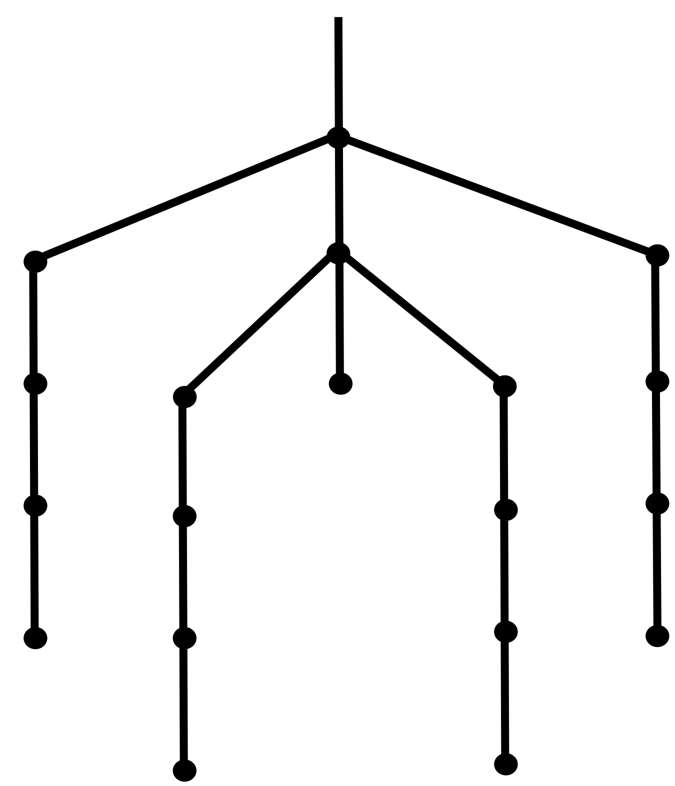
> The science of pure motion, considered without reference to the matter of objects moved, or to the force producing or changing the motion.
>
> Oxford English Dictionary

Skeletal animation

- We animate *skeletons* of characters
- Tree structure
  - Vertices are *joints*
  - Edges are *bones* (or *links*)
- A mesh is created from a skeleton every frame by *skinning*
  - A vertex is expressed as a combination of a number of positions relative to respective joints

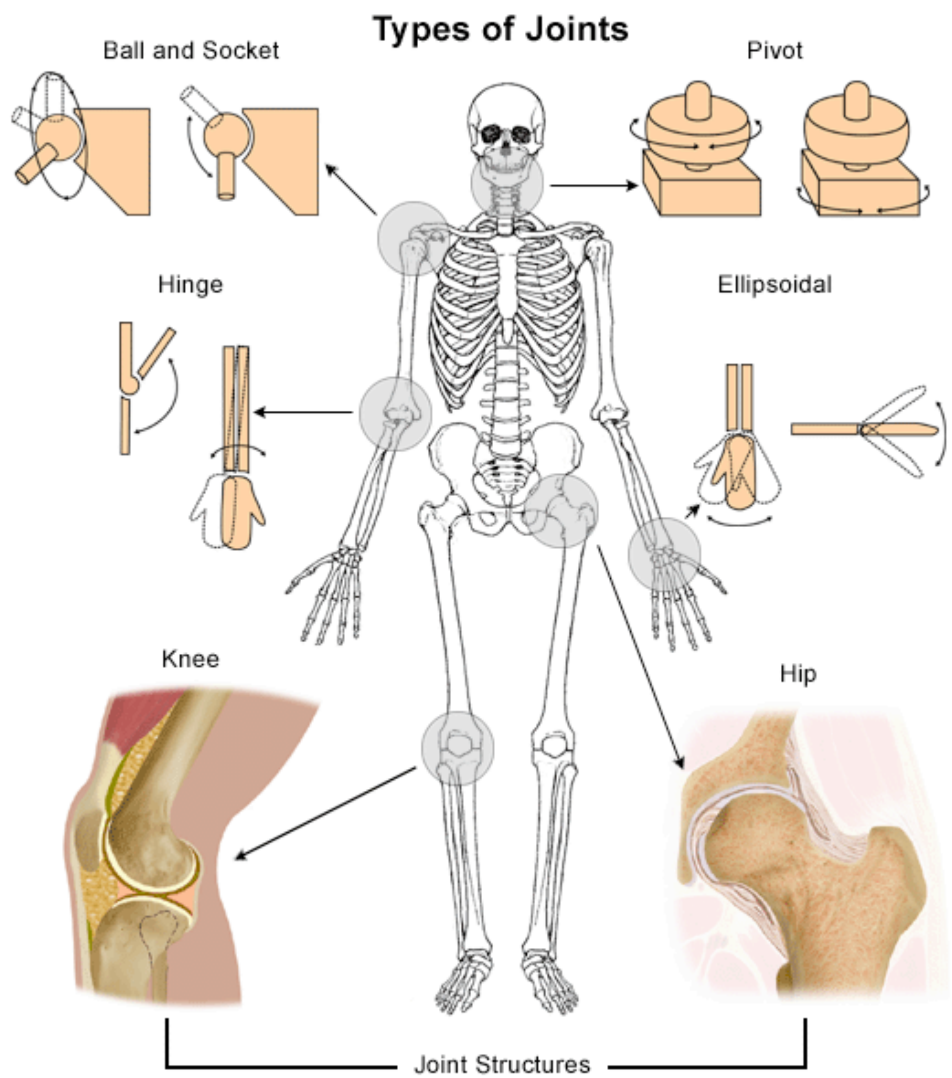
Skeletons





- Joints and bones define transformations. (Left)
- Can represent with tree structure that associates joint transformations with edges and bone transformations with vertices. (Right)

Human joints



- Human body contains over 100 types of joints, characterized by the types of induced movement.
- We represent joint configurations as rigid-body transformations.
- Types of joints correspond to different sets of allowed transformations.

Picture from http://www.cumc.columbia.edu/dept/rehab/musculoskeletal_health/anatomy.html

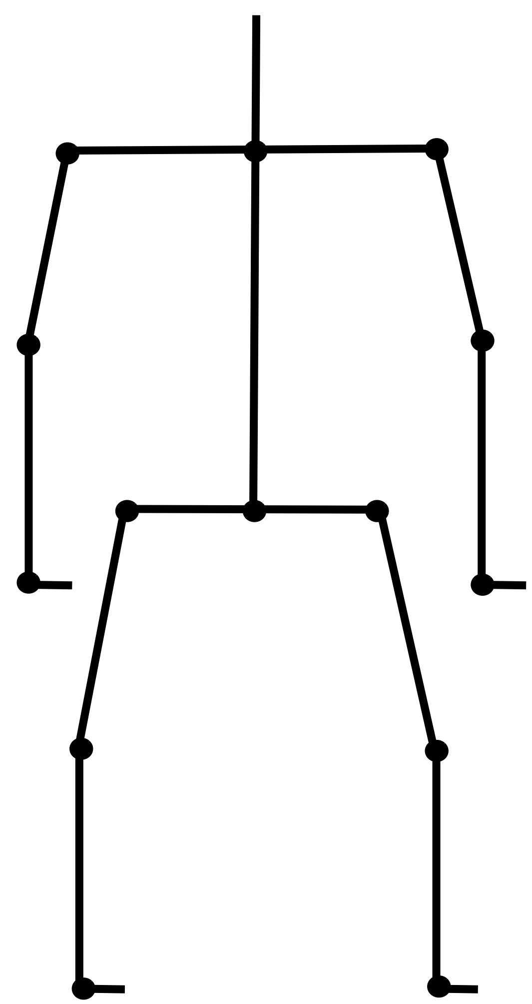
Kinematics of skeletons

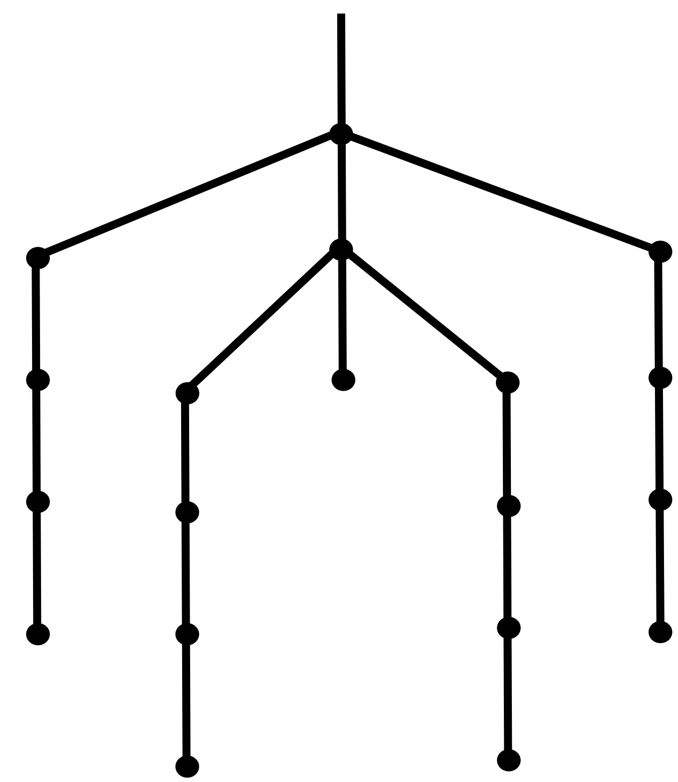
- **Forward Kinematics**: Compute configuration of a bone (*end effector*) in the global coordinate frame given configurations of all ancestor joints.
- **Inverse Kinematics**: Given configuration of a bone (*end effector*) in the global coordinate frame, compute configurations of all ancestor joints.

Kinematics of skeletons

- Forward kinematics forms a basic step for many techniques. Necessary.
- Always specifying all joint configurations manually can be tedious and unnecessary.
- Inverse kinematics used for more intuitive, higher-level specification of keys.
    - Walk character by placing feet in the right places on the ground
    - Create boxing animation by positioning fists
- In practice, animators go back and forth. (Position foot, adjust knee, rotate hip, reposition foot, adjust some more, ...)

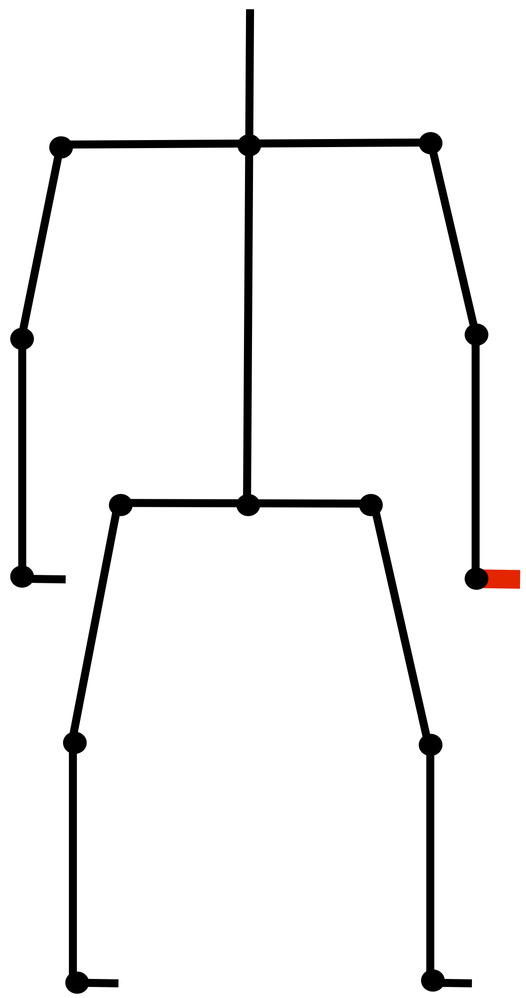
Forward Kinematics

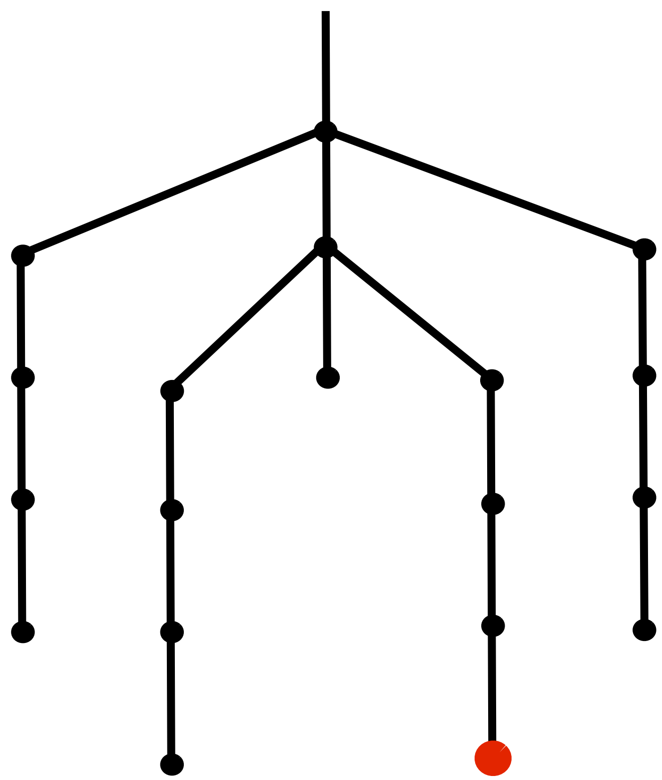




- Configuration of any bone in the global coordinate frame can be determined by tree traversal.

Forward Kinematics

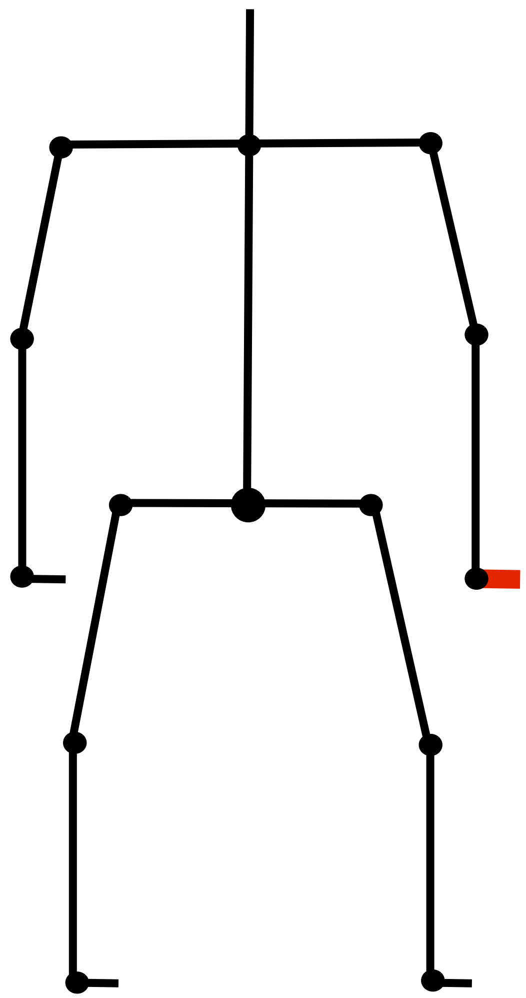


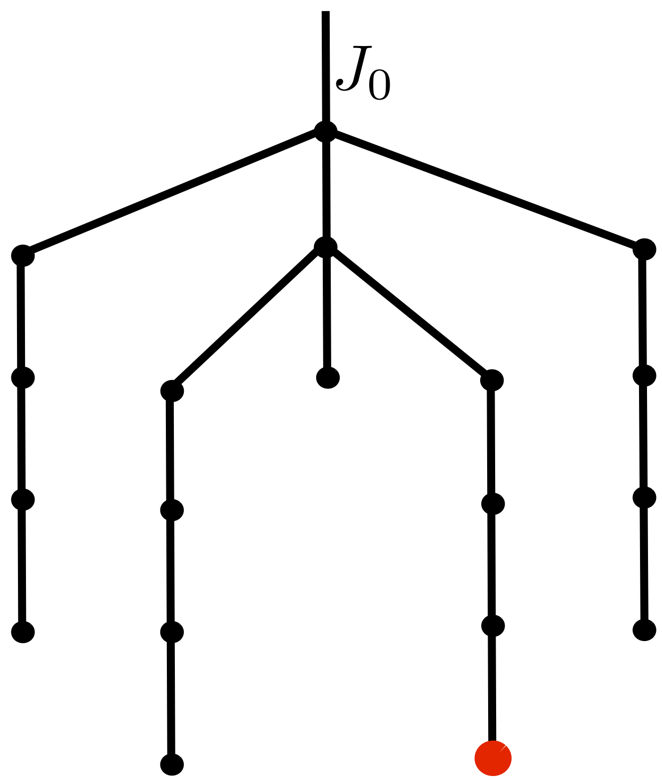


- What is the configuration of the right hand?

$$C =$$

Forward Kinematics

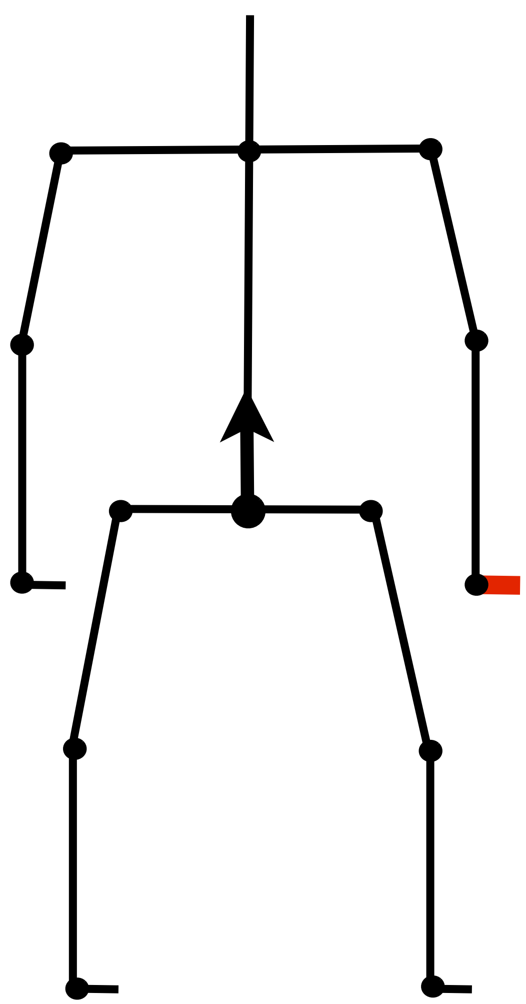


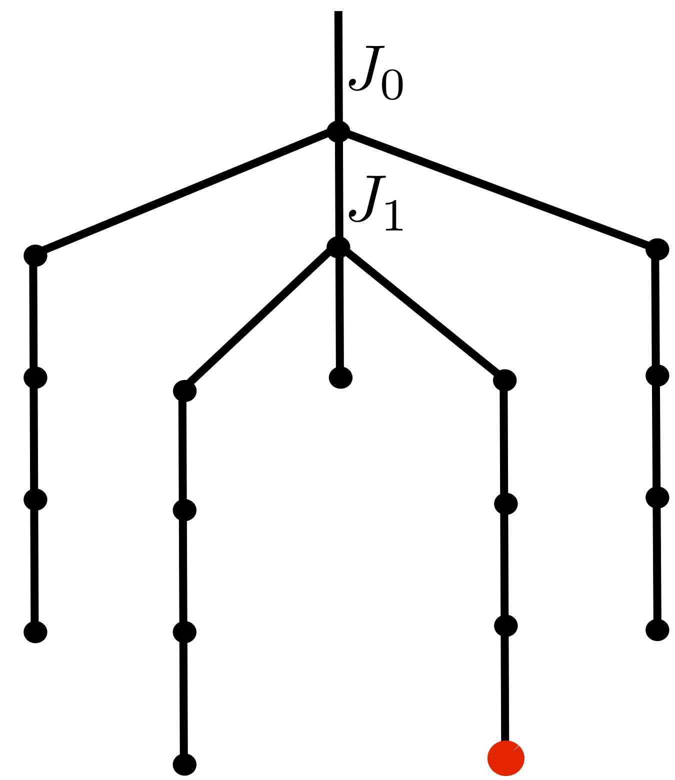


- What is the configuration of the right hand?

$$C = J_0$$

Forward Kinematics

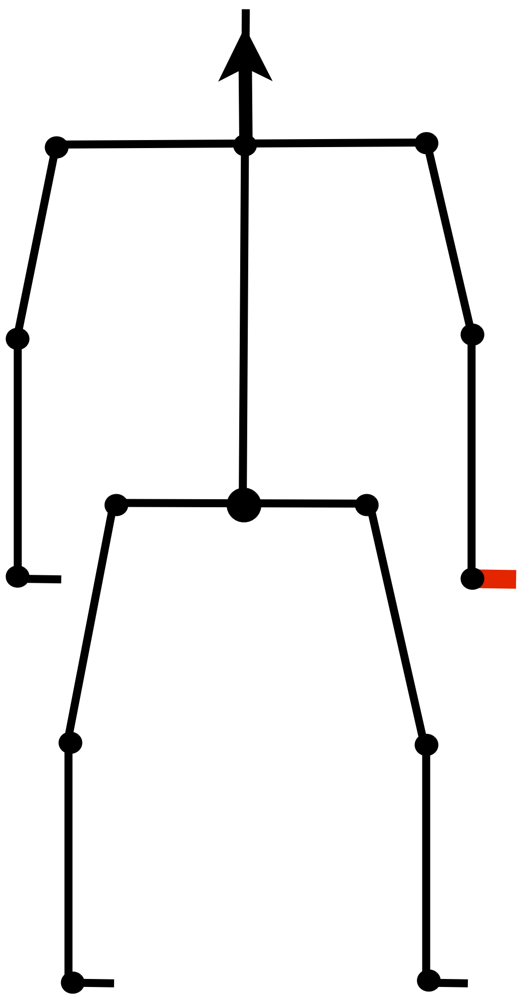


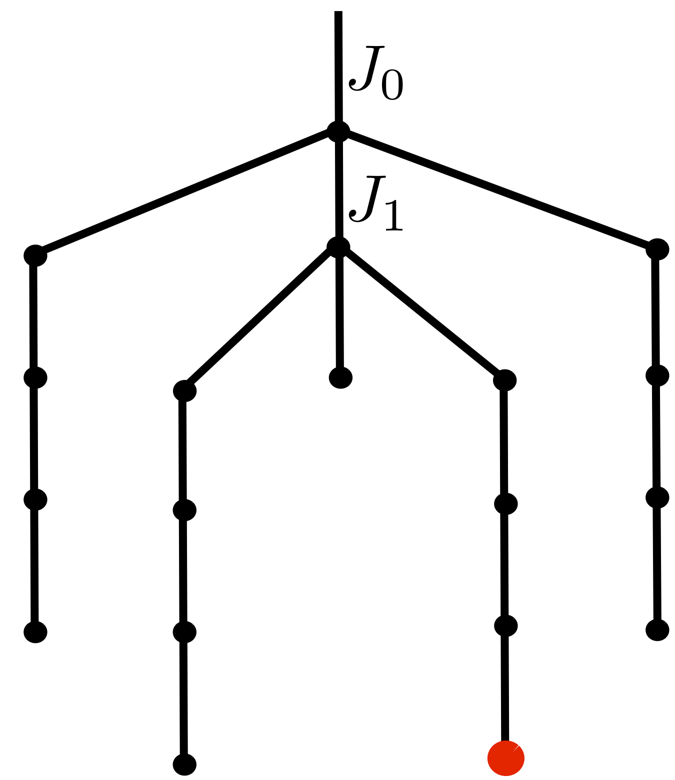


- What is the configuration of the right hand?

$$C = J_0 J_1$$

Forward Kinematics

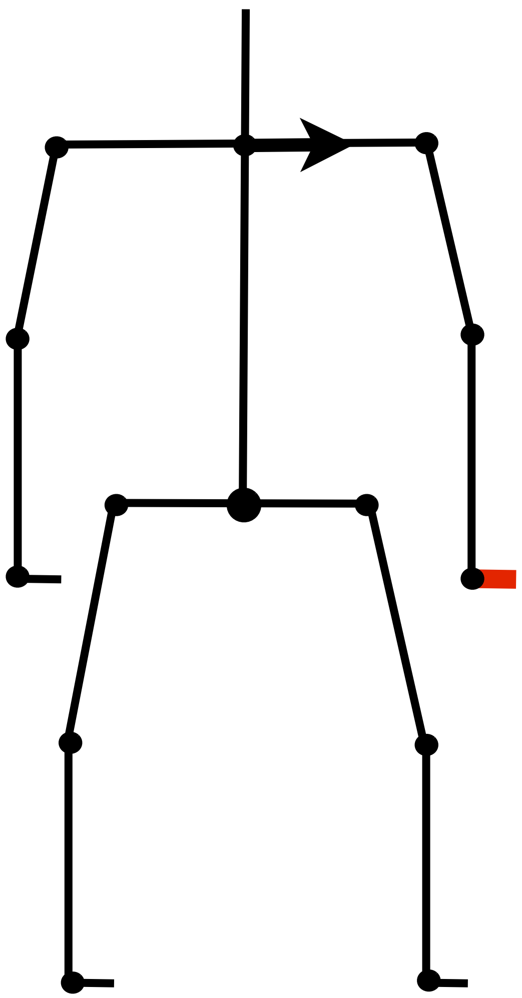


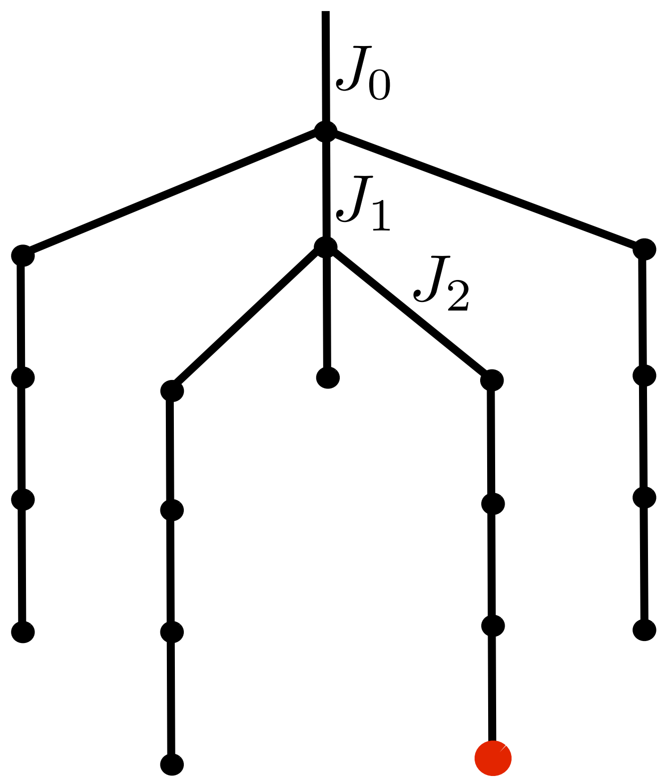


- What is the configuration of the right hand?

$$C = J_0 J_1 B_1$$

Forward Kinematics

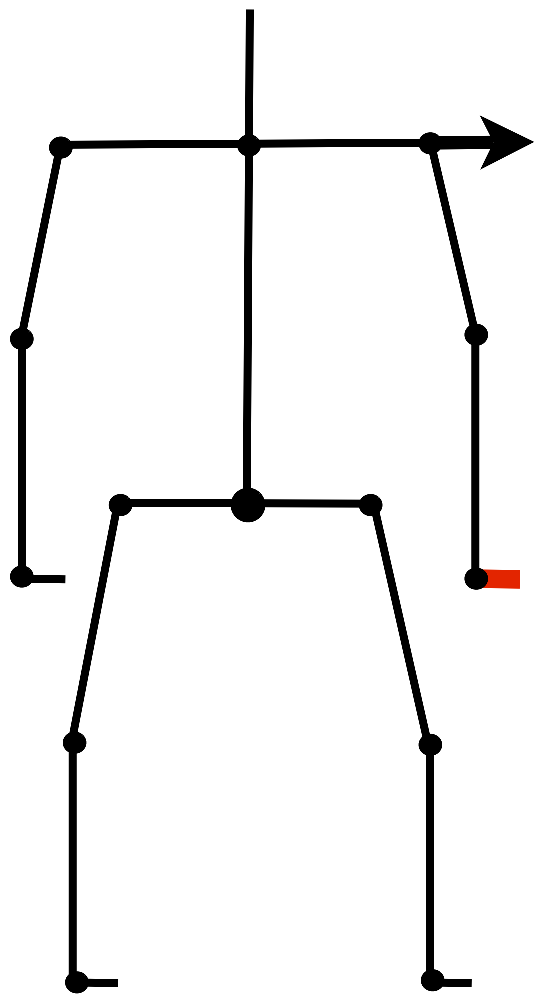


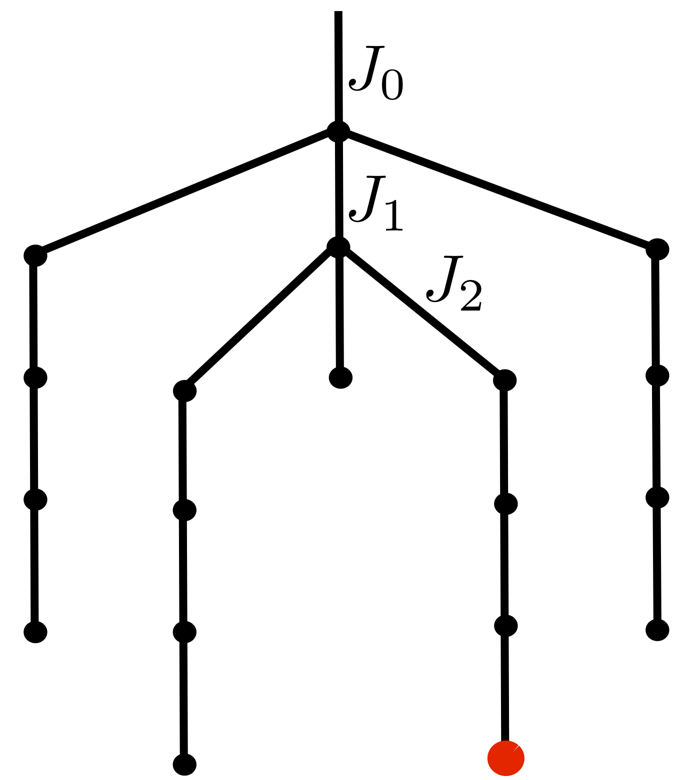


- What is the configuration of the right hand?

$$C = J_0 J_1 B_1 J_2$$

Forward Kinematics

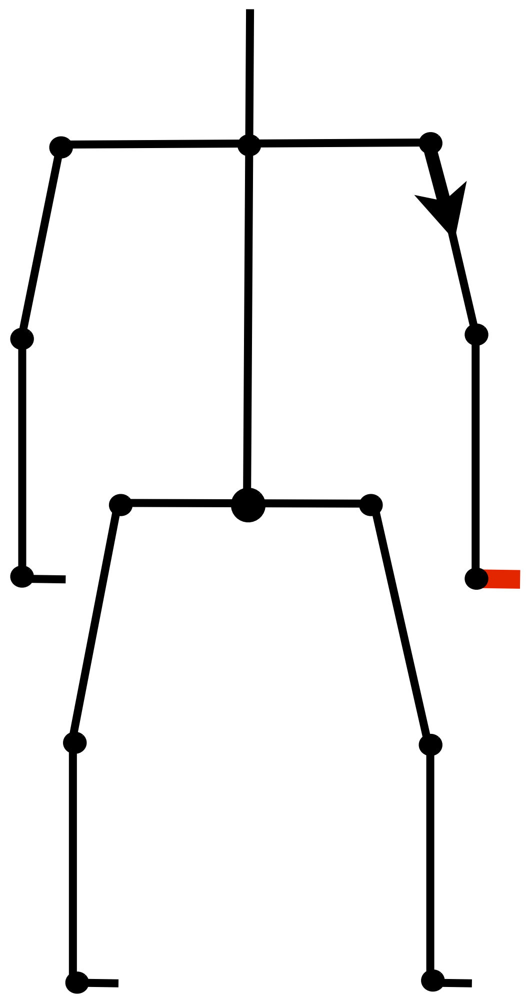


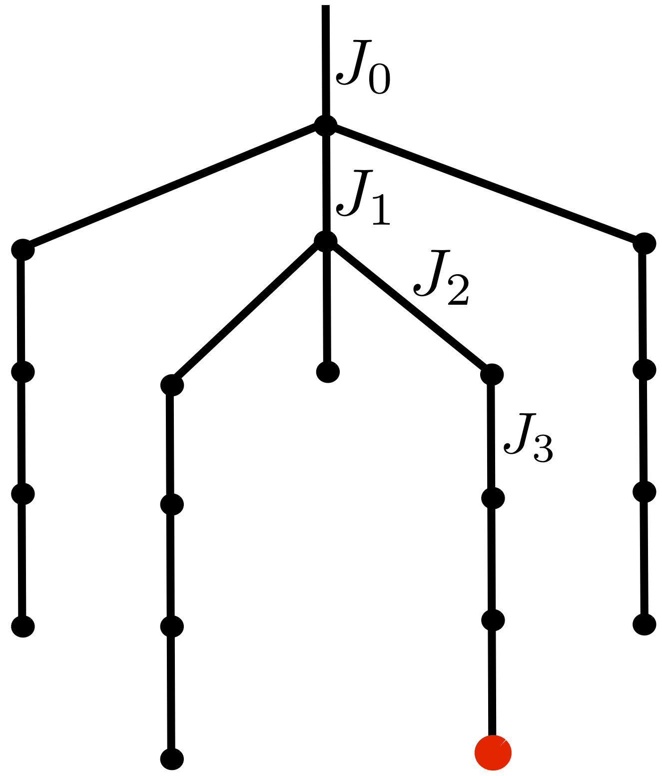


- What is the configuration of the right hand?

$$C = J_0 J_1 B_1 J_2 B_2$$

Forward Kinematics

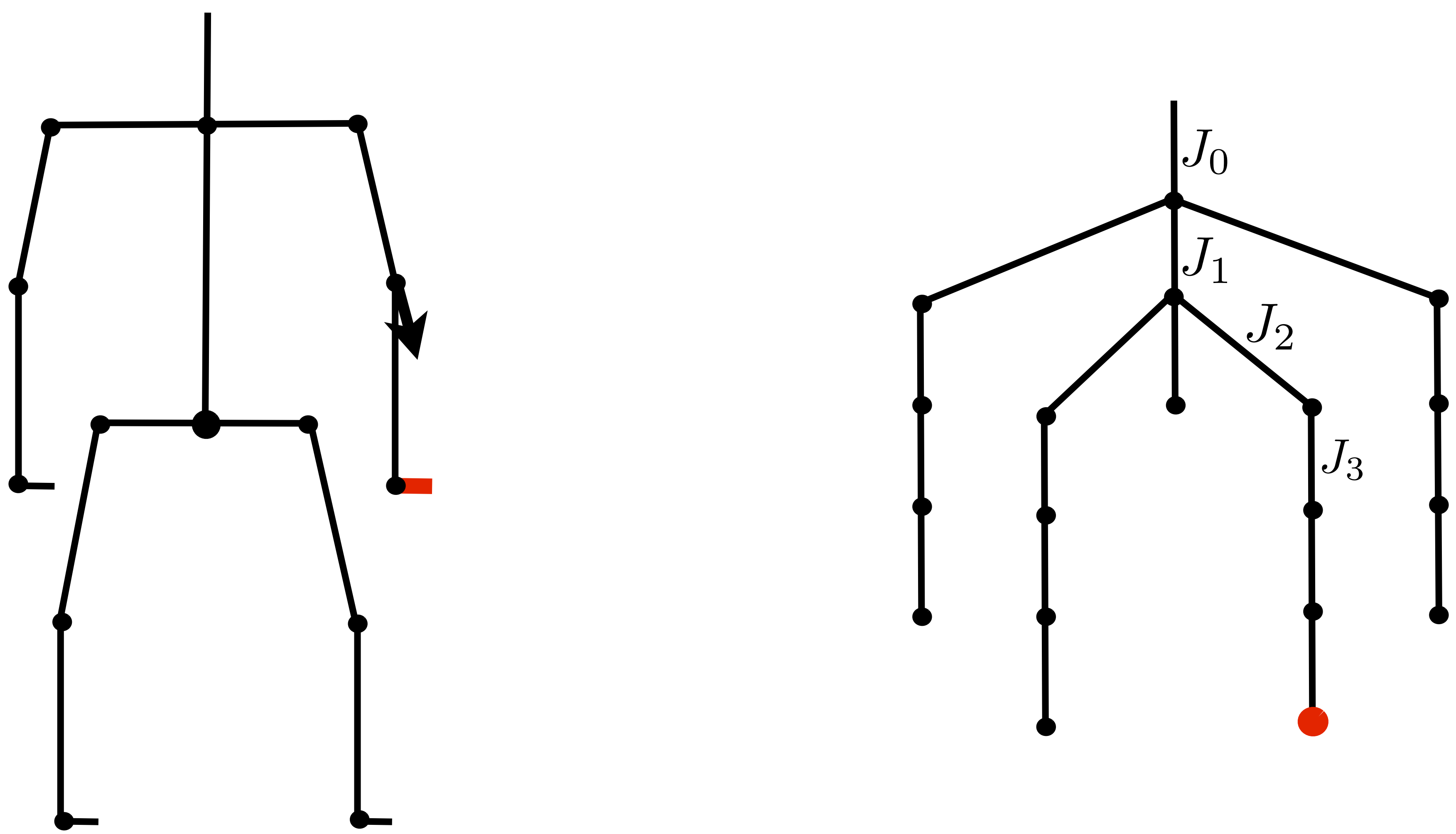




- What is the configuration of the right hand?

$$C = J_0 J_1 B_1 J_2 B_2 J_3$$

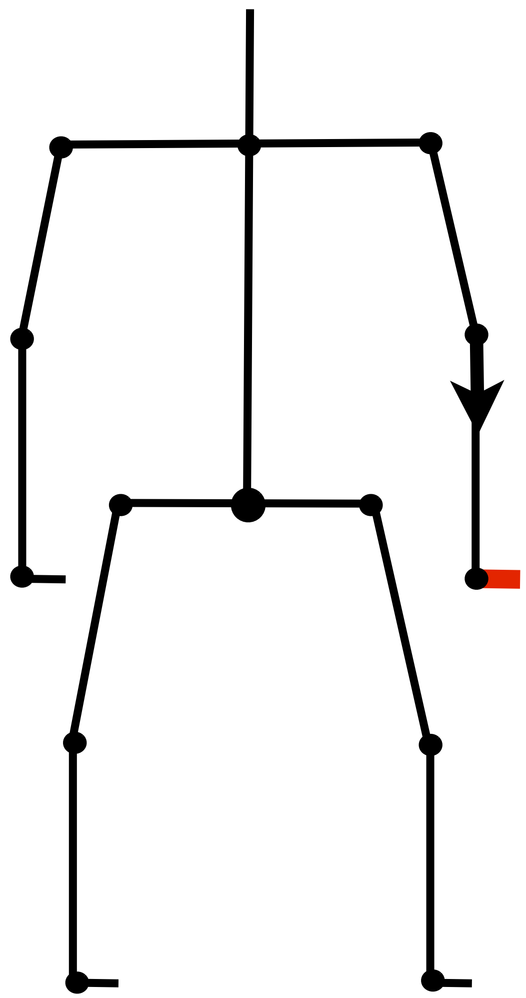
Forward Kinematics

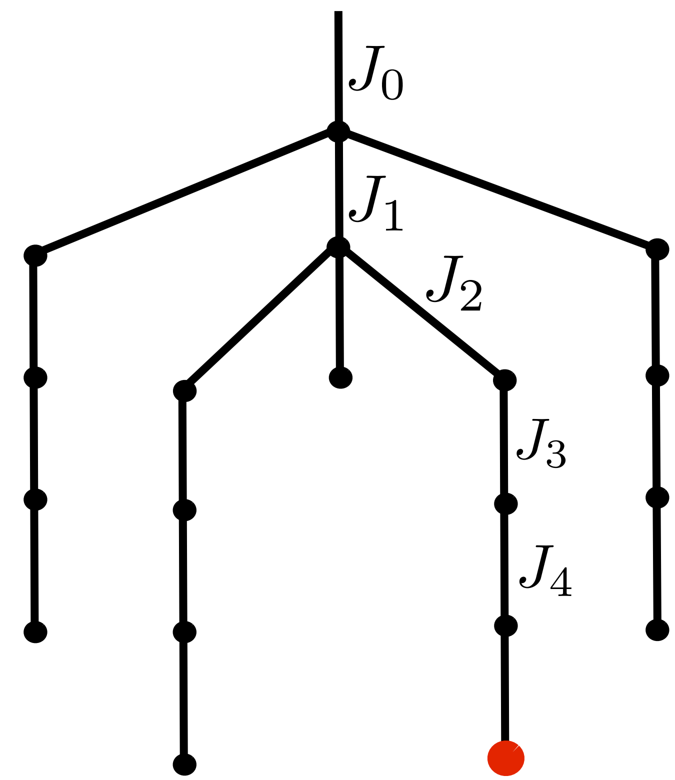


- What is the configuration of the right hand?

$$C = J_0 J_1 B_1 J_2 B_2 J_3 B_3$$

Forward Kinematics

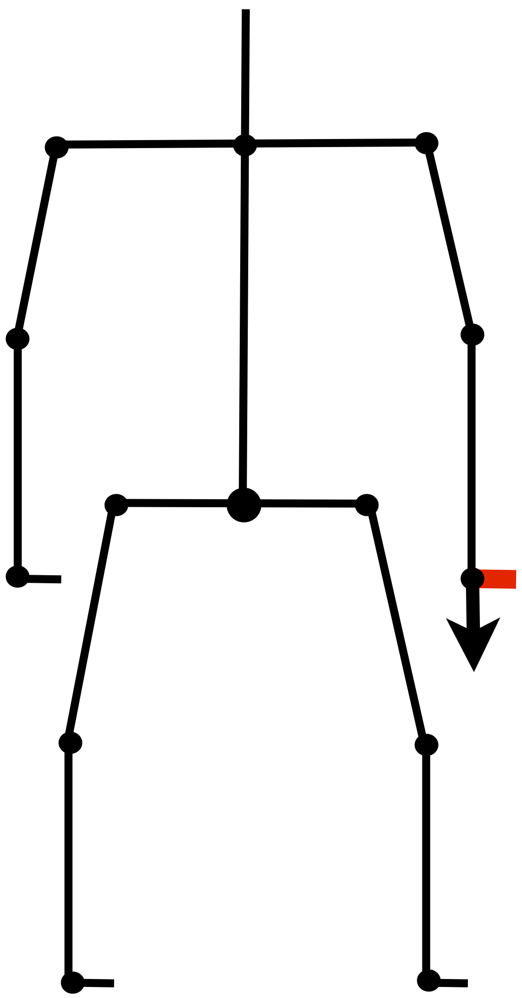


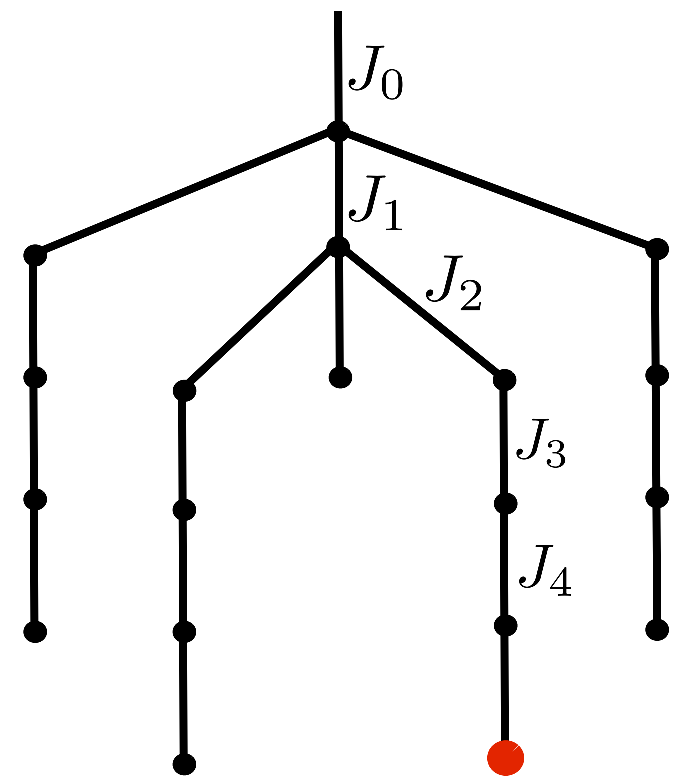


- What is the configuration of the right hand?

$$C = J_0 J_1 B_1 J_2 B_2 J_3 B_3 J_4$$

Forward Kinematics

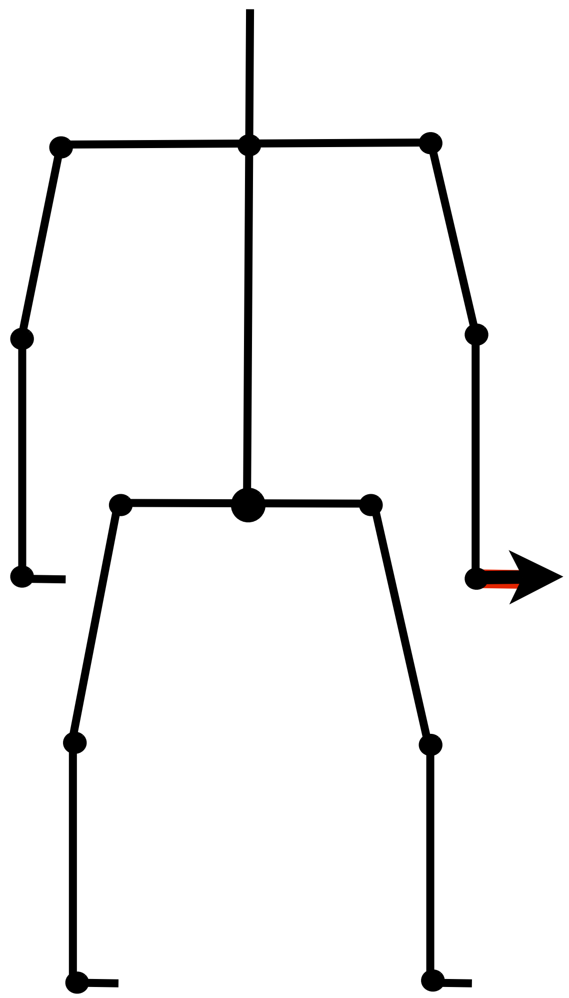


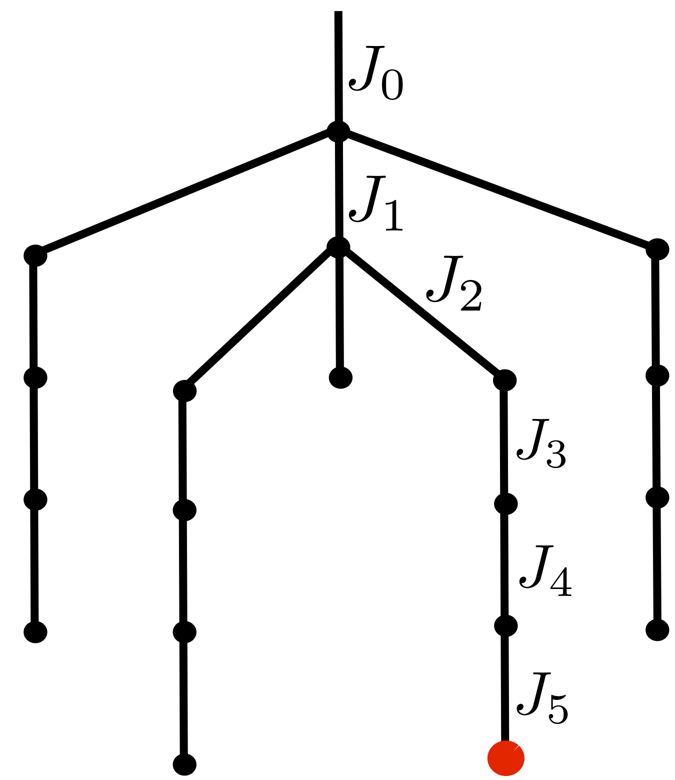


- What is the configuration of the right hand?

$$C = J_0 J_1 B_1 J_2 B_2 J_3 B_3 J_4 B_4$$

Forward Kinematics





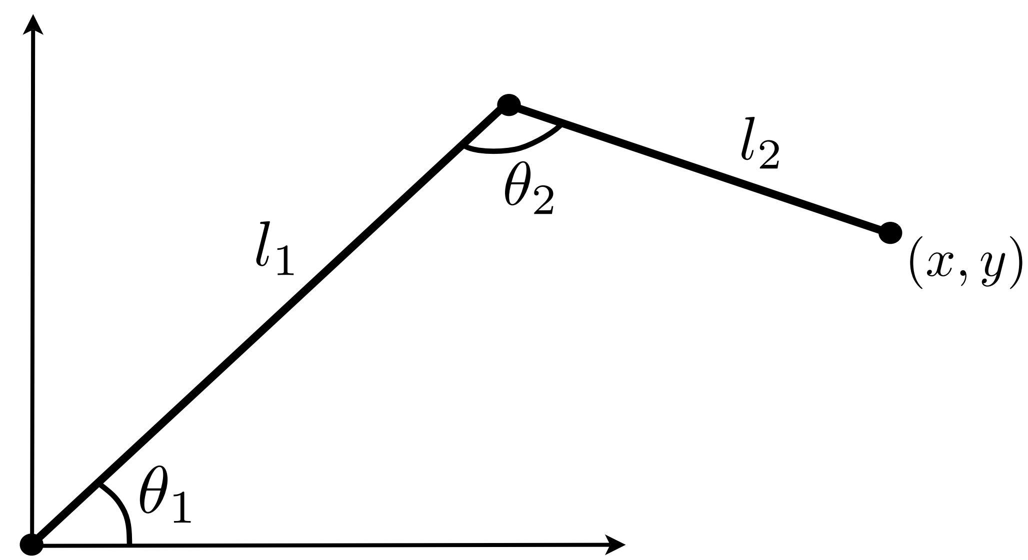
- What is the configuration of the right hand?

$$C = J_0 J_1 B_1 J_2 B_2 J_3 B_3 J_4 B_4 J_5$$

Inverse Kinematics

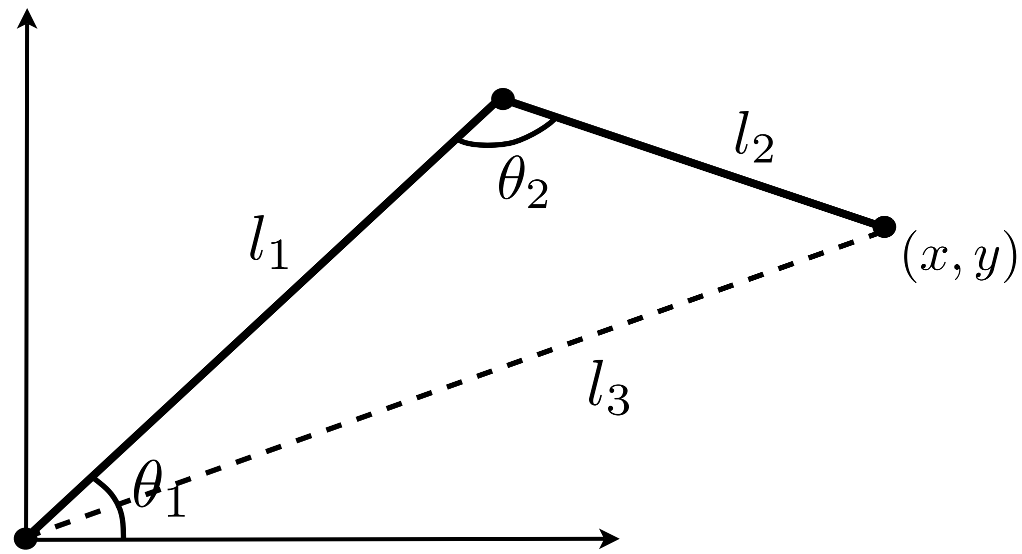
interactive demonstration

Simple example



Given l_1, l_2, and the position (x, y) of the end effector, compute θ_1 and θ_2

Simple example



$$
\begin{aligned}
l_3 &= \sqrt{x^2 + y^2} \\
\theta_2 &= \arccos \frac{l_1^2 + l_2^2 - (x^2 + y^2)}{2 l_1 l_2} \\
\theta_1 &= \arctan \frac{y}{x} + \arccos \frac{l_1^2 + (x^2 + y^2) - l_2^2}{2 l_1 \sqrt{x^2 + y^2}}
\end{aligned}
$$

General considerations

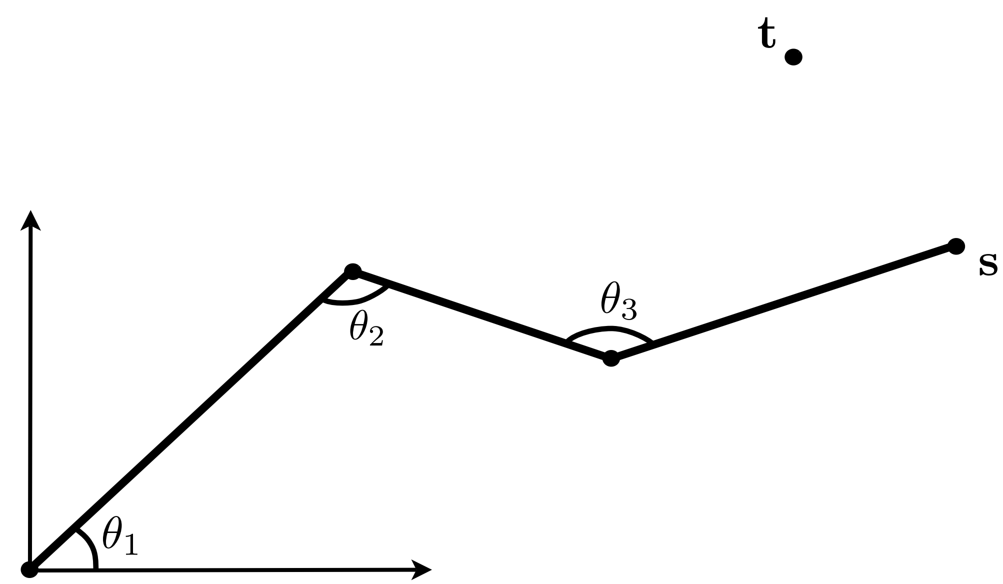
Closed-form analytical solution generally not available.

Solution might not exist. (End effector configuration not reachable.)

Number of solutions could be infinite. (Number of variables greater than number of constraints.)

We will approximate a solution iteratively. Will try to find small steps, each bringing us closer to the goal.

Iterative method

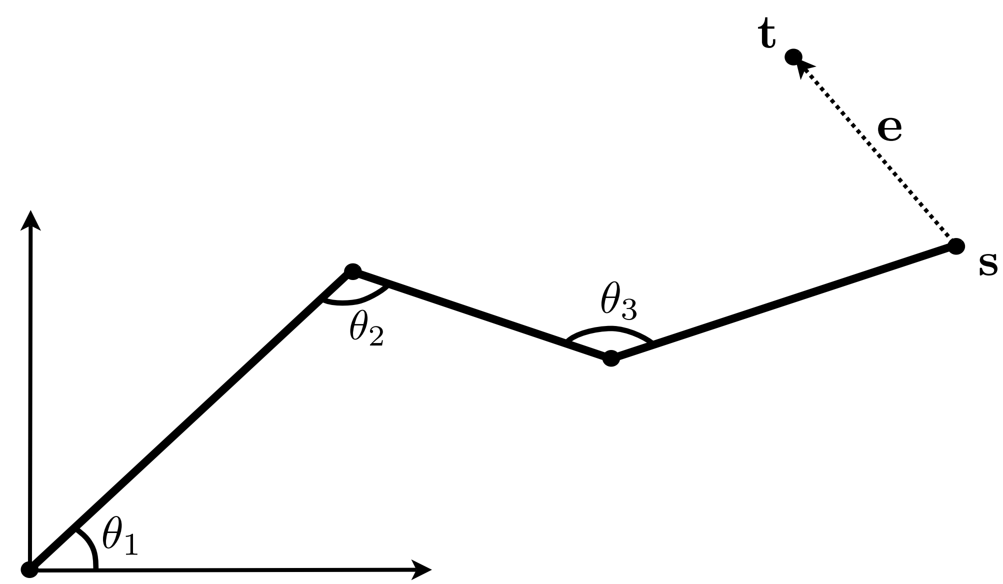


$(\theta_1, \ldots, \theta_n)$ - joint configurations (e.g., angles)

$(\mathbf{s}_1, \ldots, \mathbf{s}_k)$ - end effector configurations (e.g., positions)

$(\mathbf{t}_1, \ldots, \mathbf{t}_k)$ - target end effector configurations

Iterative method



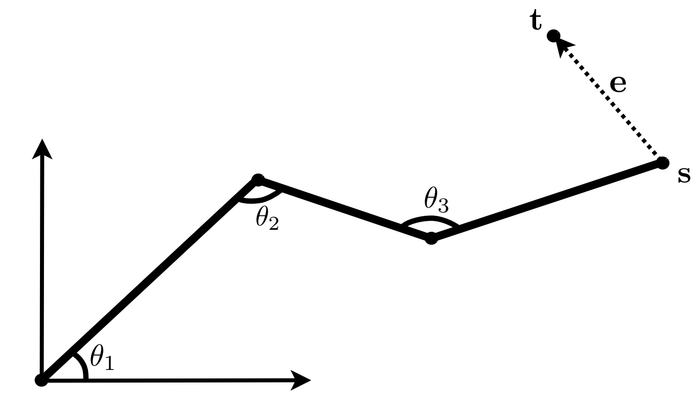
Express s as a function $\mathrm{s}(\theta)$ of θ

We want to take a step $\Delta\theta$, so that $\mathrm{s}(\theta + \Delta\theta)$ is closer to t

We will find $\Delta\theta$ through a differential $\dot{\theta}$ of a hypothetical motion $\theta(\tau)$, where τ is time

We want to find $\theta(\tau)$ that yields $\dot{\mathrm{s}} = \mathbf{e}$, where $\mathbf{e} = \mathbf{t} - \mathbf{s}$.

Iterative method



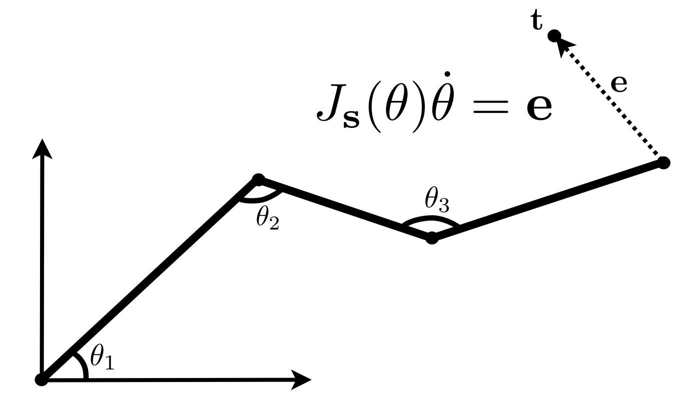
$$\dot{\mathbf{s}} = \left(\frac{d\mathbf{s}_1}{d\tau}, \ldots, \frac{d\mathbf{s}_k}{d\tau}\right)^T$$

$$\begin{aligned}\frac{d\mathbf{s}_i}{d\tau} &= \frac{d}{d\tau}\mathbf{s}_i(\theta) = \frac{d}{d\tau}\mathbf{s}_i(\theta_1(\tau), \ldots, \theta_n(\tau)) = \frac{\partial \mathbf{s}_i}{\partial \theta_1}\frac{d\theta_1}{d\tau} + \cdots + \frac{\partial \mathbf{s}_i}{\partial \theta_n}\frac{d\theta_n}{d\tau} \\ &= \left(\frac{\partial \mathbf{s}_i}{\partial \theta_1}, \ldots, \frac{\partial \mathbf{s}_i}{\partial \theta_n}\right)\left(\frac{d\theta_1}{d\tau}, \ldots, \frac{d\theta_n}{d\tau}\right)^T\end{aligned}$$

$$\begin{pmatrix} \frac{d\mathbf{s}_1}{d\tau} \\ \frac{d\mathbf{s}_2}{d\tau} \\ \vdots \\ \frac{d\mathbf{s}_k}{d\tau} \end{pmatrix} = \begin{pmatrix} \frac{\partial \mathbf{s}_1}{\partial \theta_1} & \frac{\partial \mathbf{s}_1}{\partial \theta_2} & \cdots & \frac{\partial \mathbf{s}_1}{\partial \theta_n} \\ \frac{\partial \mathbf{s}_2}{\partial \theta_1} & \frac{\partial \mathbf{s}_2}{\partial \theta_2} & \cdots & \frac{\partial \mathbf{s}_2}{\partial \theta_n} \\ \vdots & \vdots & \cdots & \vdots \\ \frac{\partial \mathbf{s}_k}{\partial \theta_1} & \frac{\partial \mathbf{s}_k}{\partial \theta_2} & \cdots & \frac{\partial \mathbf{s}_k}{\partial \theta_n} \end{pmatrix} \begin{pmatrix} \frac{d\theta_1}{d\tau} \\ \frac{d\theta_2}{d\tau} \\ \vdots \\ \frac{d\theta_n}{d\tau} \end{pmatrix}$$

$$\dot{\mathbf{s}} = J_{\mathbf{s}}(\theta)\dot{\theta}$$

Iterative method



We can compute $J_{\mathrm{s}}(\theta)$ and $\mathbf{e}$ directly. We are looking for $\dot{\theta}$ such that

$$J_{\mathbf{s}}(\theta)\dot{\theta} = \mathbf{e}$$

If $J_{\mathbf{s}}(\theta)$ is invertible, the solution is $\dot{\theta} = J_{\mathbf{s}}^{-1}(\theta)\mathbf{e}$

Otherwise the solution is given by the *pseudo-inverse*: $\dot{\theta} = J_{\mathbf{s}}^{+}(\theta)\mathbf{e}$

Let's see first how to compute $J_{\mathrm{s}}(\theta)$, and then look into the pseudo-inverse.

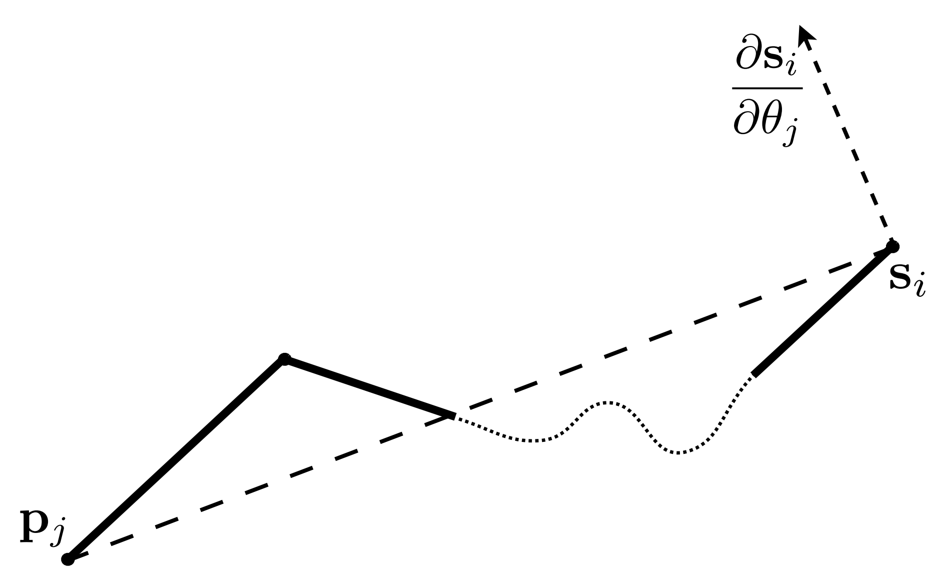
Computing the Jacobian

We consider the case where s are end-effector positions and θ are orientations of hinge joints.

$\frac{\partial \mathbf{s}_i}{\partial \theta_j}$ is the linear velocity of the rotation induced on end-effector i by hinge j. If end-effector i is unaffected by the rotation of j, the velocity is 0. Otherwise,

$$\frac{\partial \mathbf{s}_i}{\partial \theta_j} = \mathbf{v}_j \times (\mathbf{s}_i - \mathbf{p}_j)$$

where $\mathbf{p}_j$ is the position of joint j and $\mathbf{v}_j$ is the unit vector pointing along the axis of rotation according to the right hand rule.



Computing the pseudo-inverse

Need to find $\dot{\theta}$ such that $J\dot{\theta} = \mathrm{e}$.

If J has rank k (gradients for distinct end-effectors are linearly independent), then JJ^T is invertible. Consider

$$J^+ = J^T(JJ^T)^{-1} \qquad \text{and} \qquad \dot{\theta} = J^+\mathbf{e}$$

Then $$J\dot{\theta} = JJ^T(JJ^T)^{-1}\mathbf{e} = \mathbf{e}$$

Euler integration

Choose small constant α.

Update $\theta := \theta + \alpha J^{+}\mathbf{e}$

Verify that end-effector moved closer to goal. Otherwise reduce α.

Redundant degrees of freedom

When the space of joint configurations has higher dimensionality than the space of end-effector configurations, there is generally an infinite number of solutions to IK. We can try to optimize within the space of solutions to get a solution with desirable properties.

$J(I - J^+ J)\mathbf{z} = 0$ for any vector z. We set

$$\dot{\theta} = J^+\mathbf{e} + (I - J^+ J)\nabla H$$

where $H = H(\theta)$ is some function we wish to minimize. Then

$$J\dot{\theta} = \mathbf{e}$$

and every step $\theta := \theta + \alpha\big(J^+\mathbf{e} + (I - J^+ J)\nabla H\big)$

not only brings us closer to the goal end-effector configuration, but also decreases the secondary error function H.

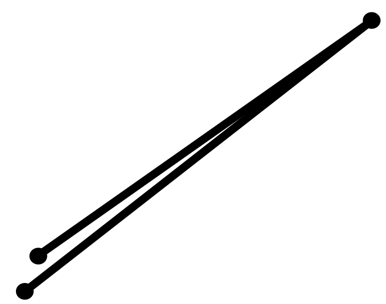
Redundant degrees of freedom

If we wish to limit joint angles to a certain range, or keep joint angles close to specific desirable values, we can set

$$H = \sum_{i=1}^{n} \beta_i (\theta_i - \theta_i^0)^2$$

where θ^0 is the optimal joint configuration and β_i are weights that act as stiffness parameters.

Singularities and Numerical Instability



Other techniques: Optimization

Can be formulated as an optimization problem in the joint configuration space.

Objective function is distance to target. A wide range of possible constraints. (Joint limits.)

At any point in the space, can evaluate objective function and its partial derivatives.