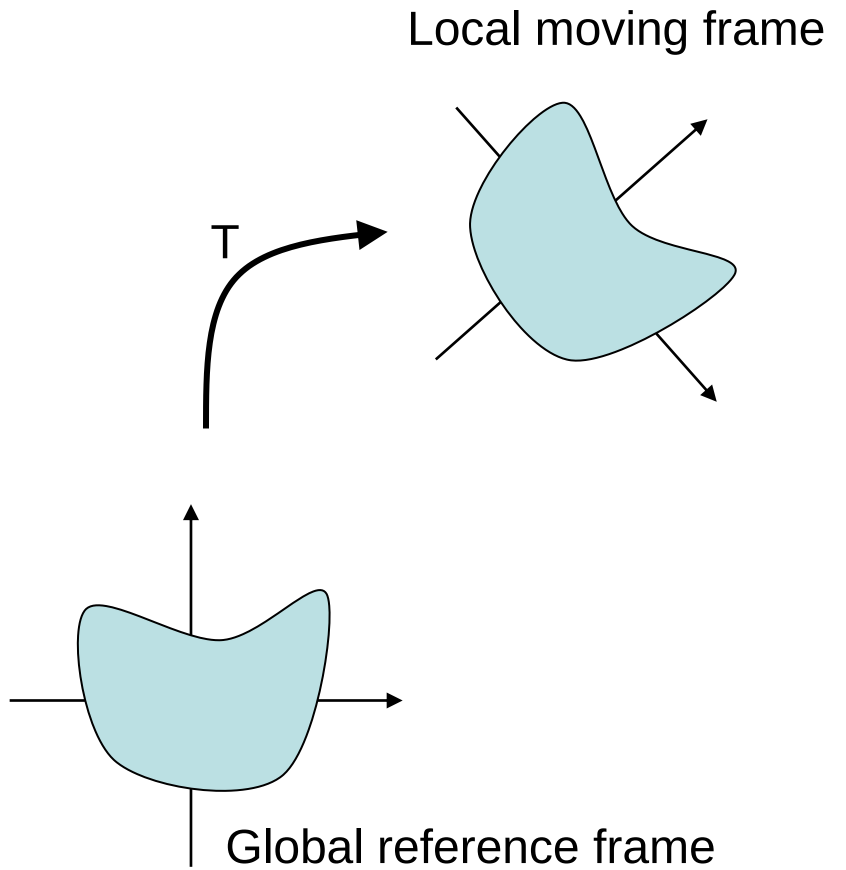
Transformations and Kinematics

Jehee Lee
Seoul National University

Transformations

- Linear transformations
- Rigid transformations
- Affine transformations
- Projective transformations



Linear Transformations

- A ***linear transformation*** T is a mapping between vector spaces
  - T maps vectors to vectors
  - linear combination is invariant under T

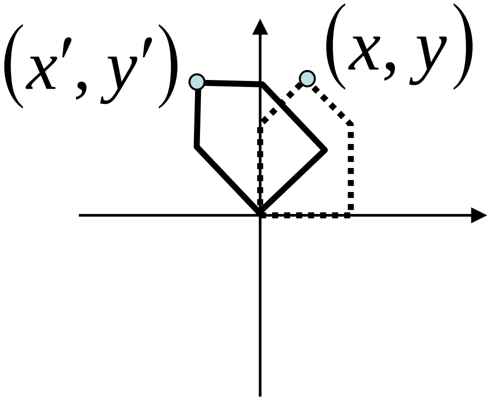
$$T\left(\sum_{i=0}^{N} c_i \mathbf{v}_i\right) = c_0 T(\mathbf{v}_0) + c_1 T(\mathbf{v}_1) + \cdots + c_N T(\mathbf{v}_N)$$

- In 3-spaces, T can be represented by a 3x3 matrix

$$\begin{aligned} T(\mathbf{v}) &= \mathbf{M}_{3\times3}\mathbf{v}_{3\times1} \quad \text{(Column major)} \\ &= \mathbf{v}_{1\times3}\mathbf{N}_{3\times3} \quad \text{(Row major)} \end{aligned}$$

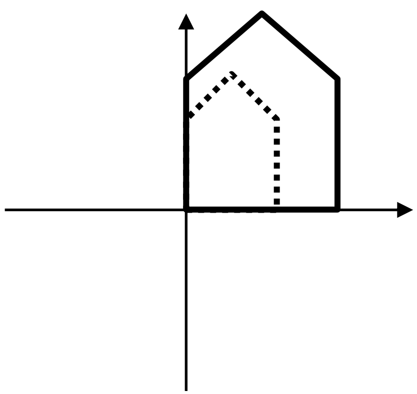
Examples of Linear Transformations

- 2D rotation



$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}$$

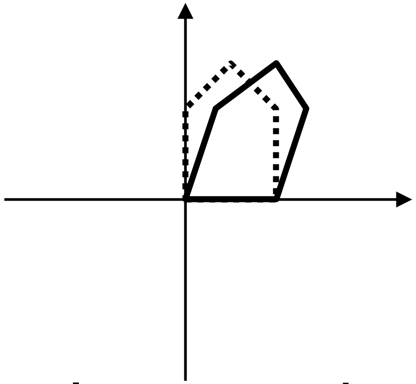
- 2D scaling



$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} s_x & 0 \\ 0 & s_y \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} s_x x \\ s_y y \end{pmatrix}$$

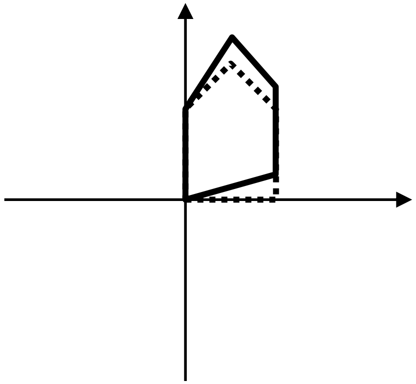
Examples of Linear Transformations

- 2D shear
  - Along X-axis



$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} 1 & d \\ 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} x + dy \\ y \end{pmatrix}$$

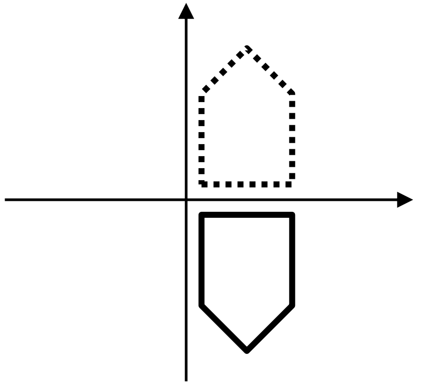
  - Along Y-axis



$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ d & 1 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} x \\ y + dx \end{pmatrix}$$

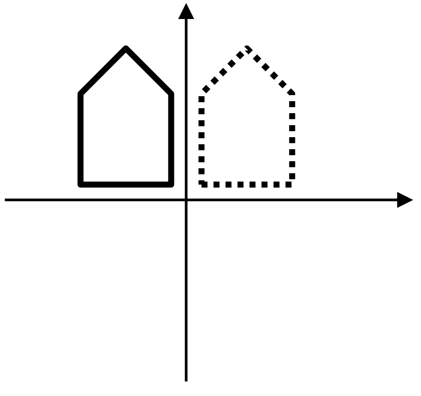
Examples of Linear Transformations

- 2D reflection
  - Along X-axis



$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} x \\ -y \end{pmatrix}$$

  - Along Y-axis



$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} -x \\ y \end{pmatrix}$$

Properties of Linear Transformations

- Any ***linear transformation*** between 3D spaces can be represented by a 3x3 matrix
- Any ***linear transformation*** between 3D spaces can be represented as a combination of ***rotation***, ***shear***, and ***scaling***
- ***Rotation*** can be represented as a combination of ***scaling*** and ***shear***

Affine Transformations

- An ***affine transformation*** T is an mapping between affine spaces
  - T maps vectors to vectors, and points to points
  - T is a linear transformation on vectors
  - affine combination is invariant under T

$$T\left(\sum_{i=0}^{N} c_i \mathbf{p}_i\right) = c_0 T(\mathbf{p}_0) + c_1 T(\mathbf{p}_1) + \cdots + c_N T(\mathbf{p}_N)$$

- In 3-spaces, T can be represented by a 3x3 matrix together with a 3x1 translation vector

$$T(\mathbf{p}) = \mathbf{M}_{3\times3}\mathbf{p}_{3\times1} + \mathbf{T}_{3\times1}$$

Homogeneous Coordinates

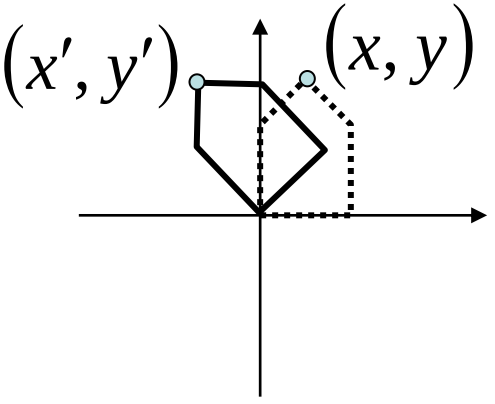
- Any affine transformation between 3D spaces can be represented by a 4x4 matrix

$$T(\mathbf{p}) = \begin{pmatrix} \mathbf{M}_{3\times3} & \mathbf{T}_{3\times1} \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \mathbf{p}_{3\times1} \\ 1 \end{pmatrix}$$

- Affine transformation is *linear* in homogeneous coordinates

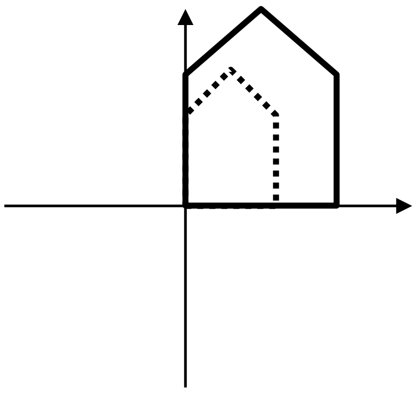
Examples of Affine Transformations

- 2D rotation



$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

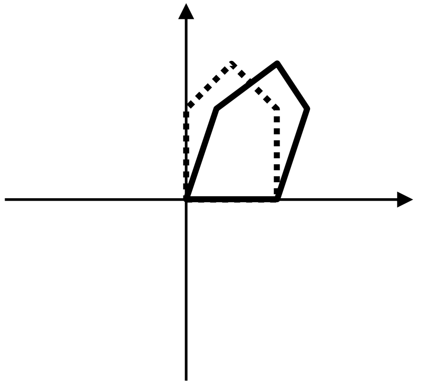
- 2D scaling



$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \begin{pmatrix} s_x x \\ s_y y \\ 1 \end{pmatrix}$$

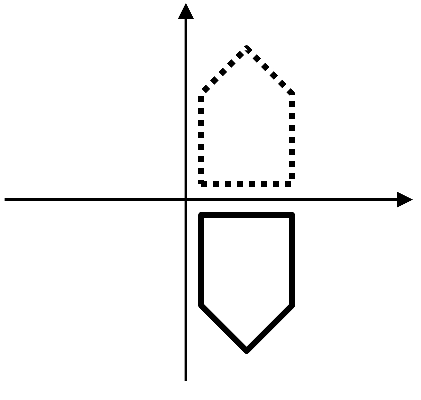
Examples of Affine Transformations

- 2D shear



$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & d & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \begin{pmatrix} x + dy \\ y \\ 1 \end{pmatrix}$$

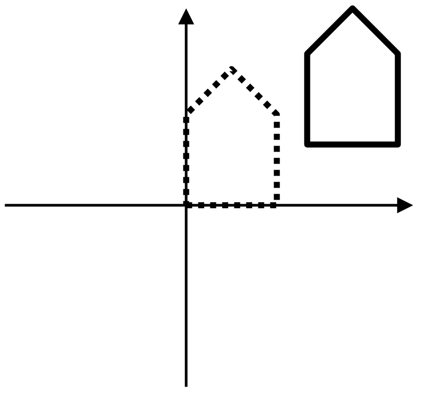
- 2D reflection



$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \begin{pmatrix} x \\ -y \\ 1 \end{pmatrix}$$

Examples of Affine Transformations

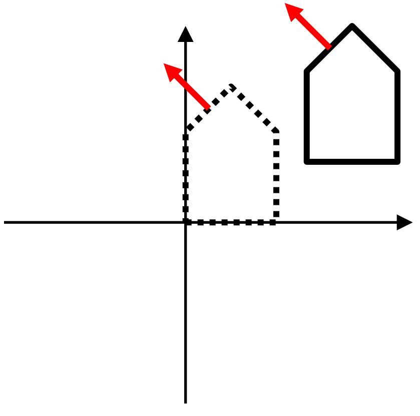
- 2D translation



$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \begin{pmatrix} x + t_x \\ y + t_y \\ 1 \end{pmatrix}$$

Examples of Affine Transformations

- 2D transformation for ***vectors***
  - ***Translation*** is simply ignored



$$\begin{pmatrix} x' \\ y' \\ 0 \end{pmatrix} = \begin{pmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 0 \end{pmatrix} = \begin{pmatrix} x \\ y \\ 0 \end{pmatrix}$$

Properties of Affine Transformations

- Any ***affine transformation*** between 3D spaces can be represented as a combination of a ***linear transformation*** followed by ***translation***
- An affine transf. maps ***lines*** to ***lines***
- An affine transf. maps ***parallel lines*** to ***parallel lines***
- An affine transf. preserves ***ratios of distance*** along a line
- An affine transf. does not preserve absolute distances and angles

Rigid Transformations

- A ***rigid transformation*** T is a mapping between affine spaces
  - T maps vectors to vectors, and points to points
  - T preserves distances between all points
  - T preserves cross product for all vectors (to avoid reflection)
- In 3-spaces, T can be represented as

$$T(\mathbf{p}) = \mathbf{R}_{3\times3}\mathbf{p}_{3\times1} + \mathbf{T}_{3\times1}, \quad \text{where}$$

$$\mathbf{R}\mathbf{R}^T = \mathbf{R}^T\mathbf{R} = \mathbf{I} \quad \text{and} \quad \det \mathbf{R} = 1$$

Rigid Body Rotation

- Rigid body transformations allow only rotation and translation
- Rotation matrices form SO(3)
  - Special orthogonal group

$$\mathbf{R}\mathbf{R}^T = \mathbf{R}^T\mathbf{R} = \mathbf{I}$$ (Distance preserving)

$$\det \mathbf{R} = 1$$ (No reflection)

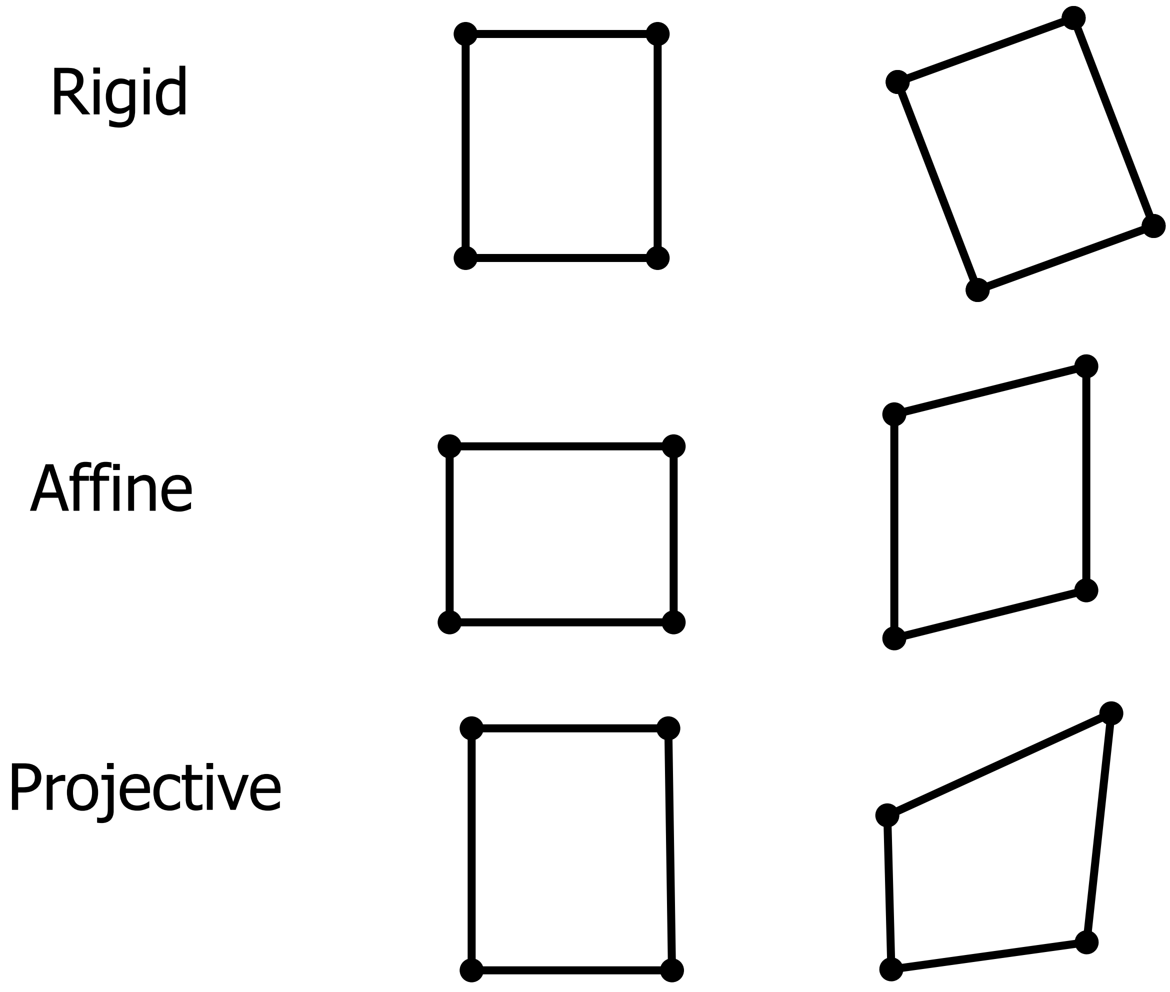
Rigid Body Rotation

- R is normalized
  - The squares of the elements in any row or column sum to 1
- R is orthogonal $\mathbf{R}\mathbf{R}^T = \mathbf{R}^T\mathbf{R} = \mathbf{I}$
  - The dot product of any pair of rows or any pair columns is 0
- The rows (columns) of R correspond to the vectors of the principle axes of the rotated coordinate frame

Taxonomy of Transformations

- **Linear** transformations
  - 3x3 matrix
  - Rotation + scaling + shear
- **Rigid** transformations
  - SO(3) for rotation
  - 3D vector for translation
- **Affine** transformation
  - 3x3 matrix + 3D vector or 4x4 homogenous matrix
  - Linear transformation + translation
- **Projective** transformation
  - 4x4 matrix
  - Affine transformation + perspective projection

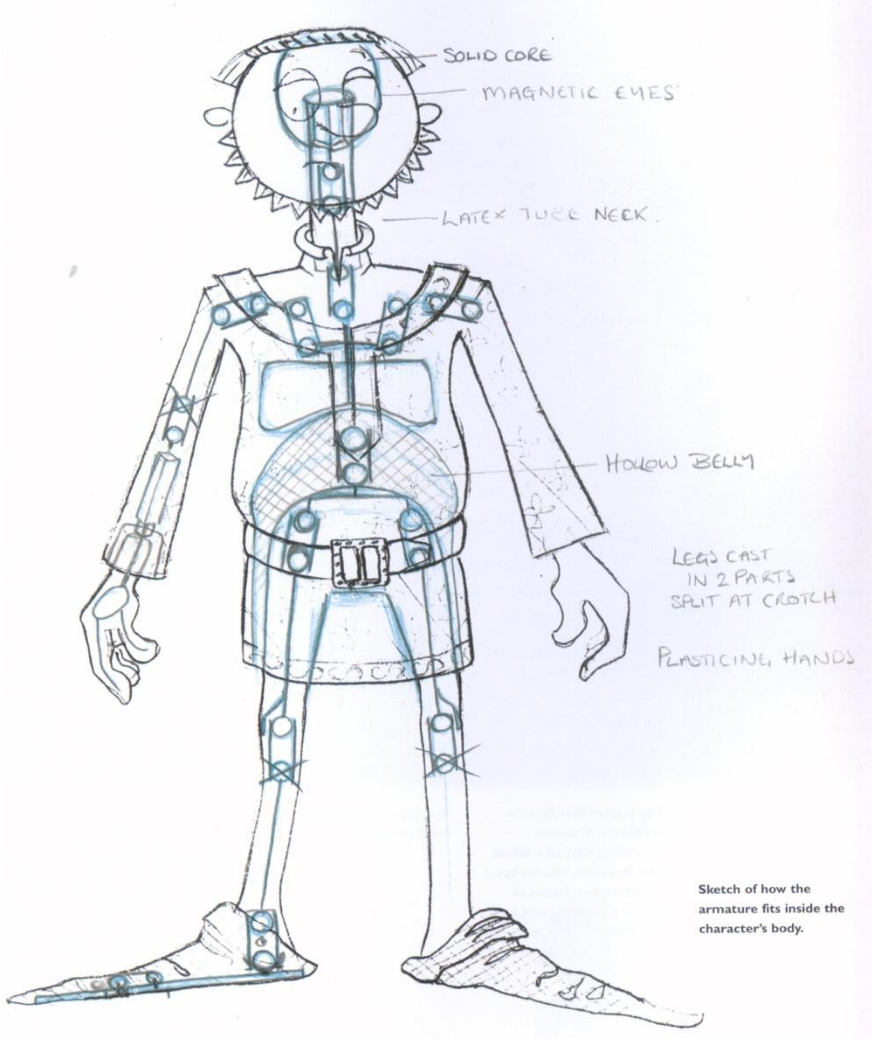
Taxonomy of Transformations



Kinematics

- How to animate skeletons (articulated figures)

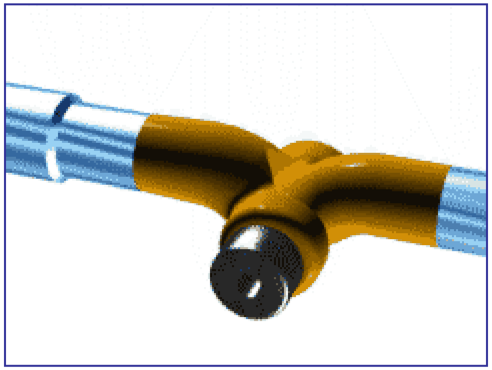
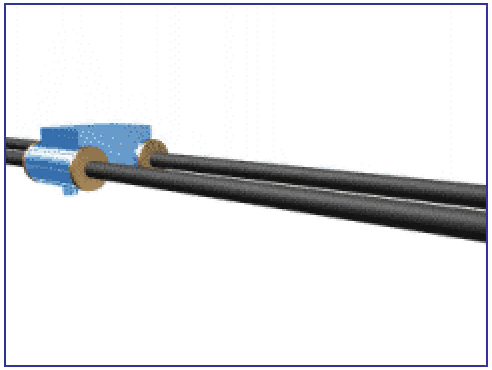
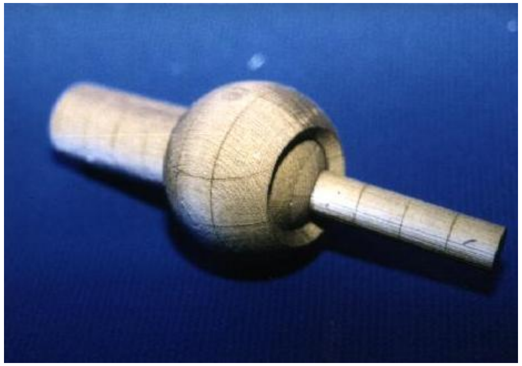
- ***Kinematics*** is the study of motion without regard to the forces that caused it



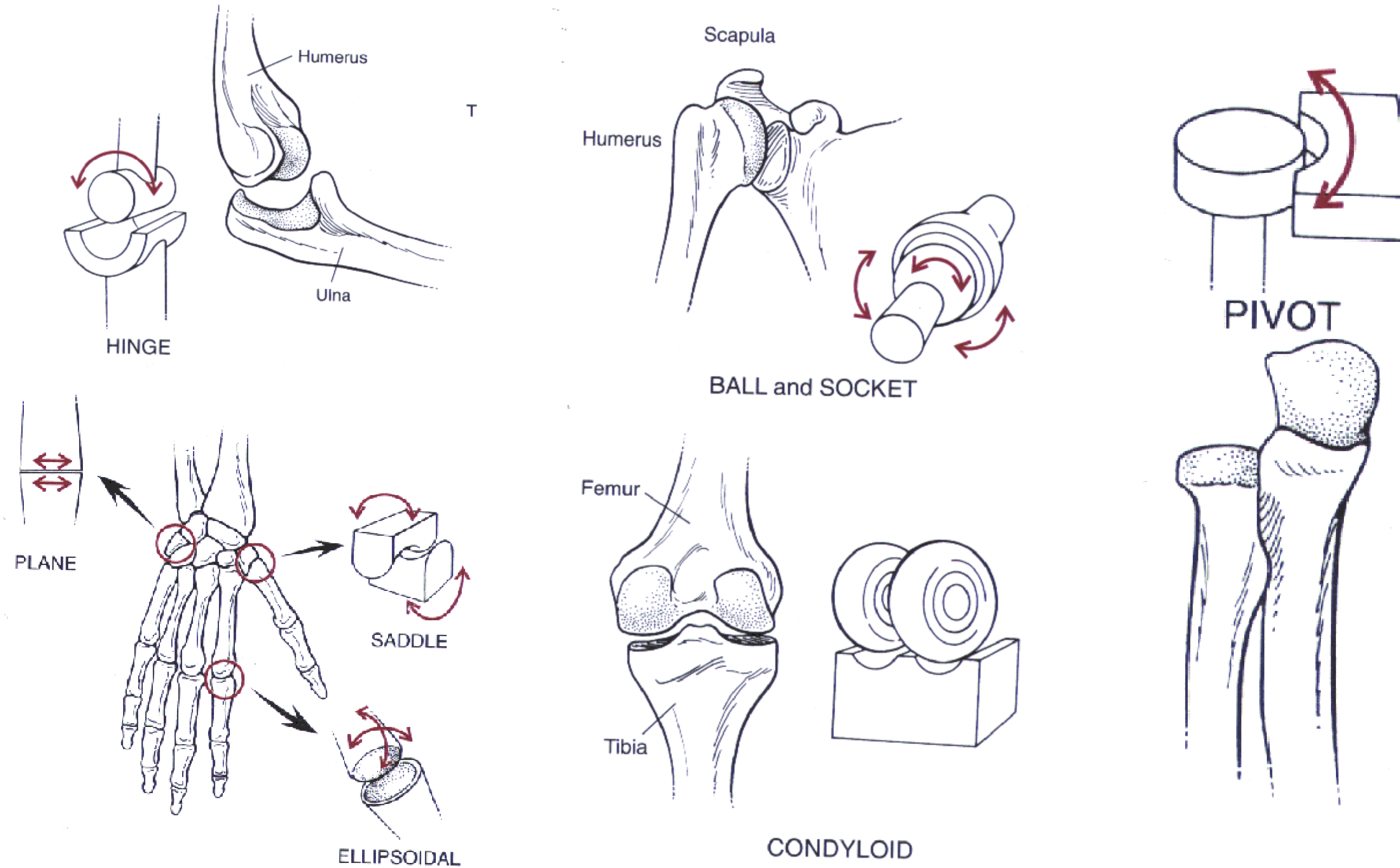
Sketch of how the armature fits inside the character's body.

Hierarchical Models

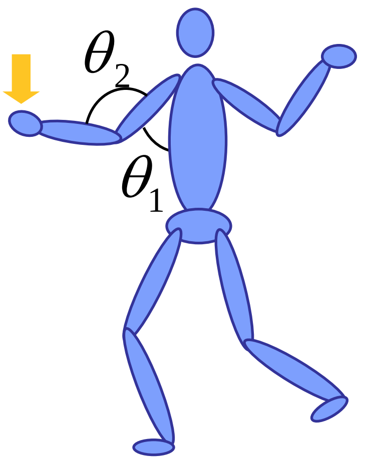
- Tree structure of joints and links
  - The root link can be chosen arbitrarily
- Joints
  - Revolute (hinge) joint allows rotation about a fixed axis
  - Prismatic joint allows translation along a line
  - Ball-and-socket joint allows rotation about an arbitrary axis

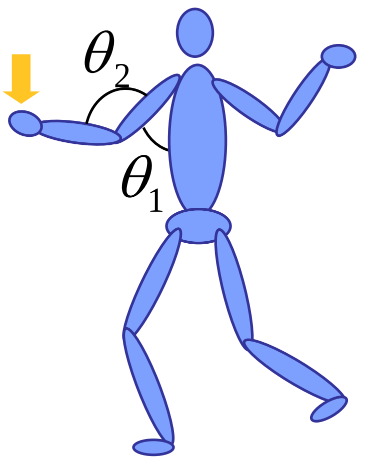




Human Joints



Forward and Inverse Kinematics





$$(\mathbf{p}, \mathbf{q}) = \mathrm{F}(\theta_i)$$

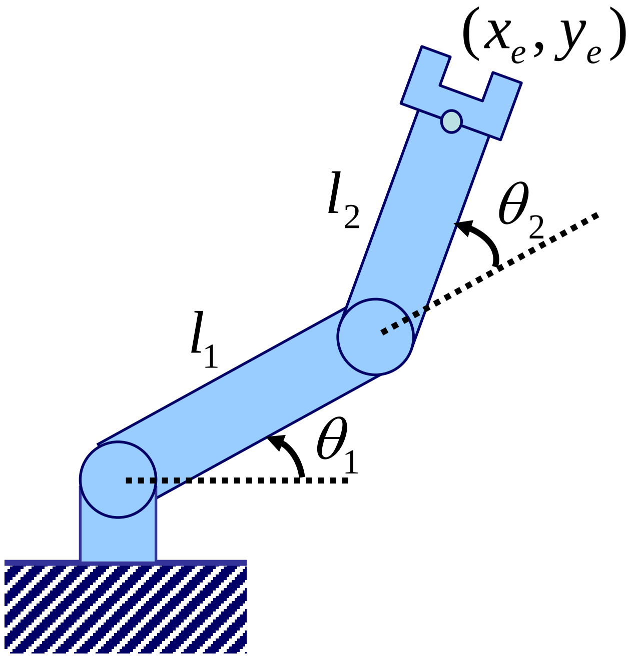
Forward Kinematics

$$\theta_i = \mathrm{F}^{-1}(\mathbf{p}, \mathbf{q})$$

Inverse Kinematics

Forward Kinematics: A Simple Example

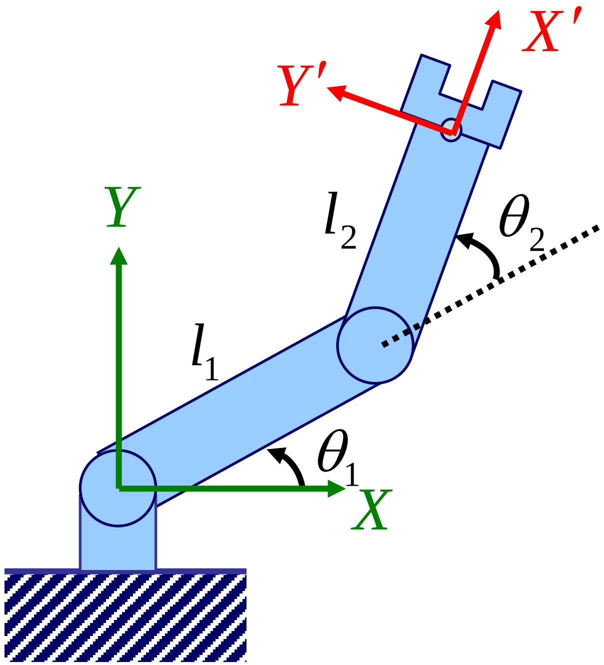
- A simple robot arm in 2-dimensional space
  - 2 revolute joints
  - Joint angles are known
  - Compute the position of the end-effector



$$x_e = l_1 \cos\theta_1 + l_2 \cos(\theta_1 + \theta_2)$$
$$y_e = l_1 \sin\theta_1 + l_2 \sin(\theta_1 + \theta_2)$$

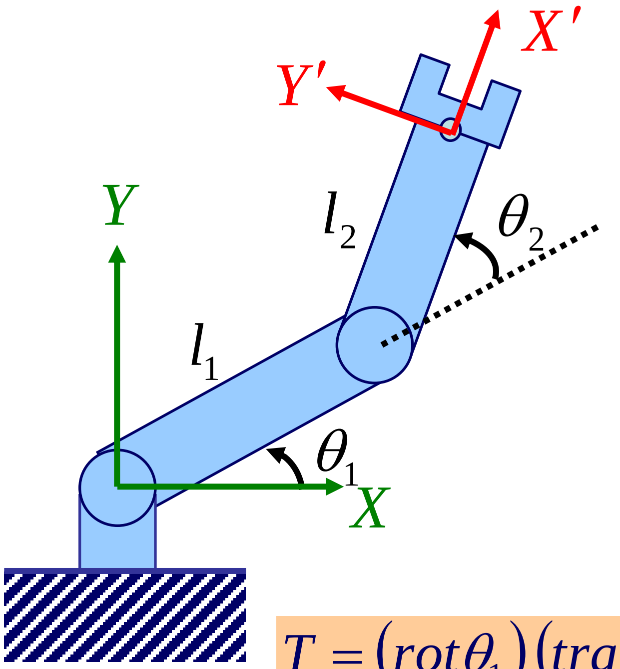
Forward Kinematics: A Simple Example

- Forward kinematics map as a coordinate transformation
  - The body local coordinate system of the end-effector was initially coincide with the global coordinate system
  - Forward kinematics map transforms the position and orientation of the end-effector according to joint angles



$$\begin{pmatrix} x_e \\ y_e \\ 1 \end{pmatrix} = \begin{pmatrix} & & \\ & T & \\ & & \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$$

A Chain of Transformations



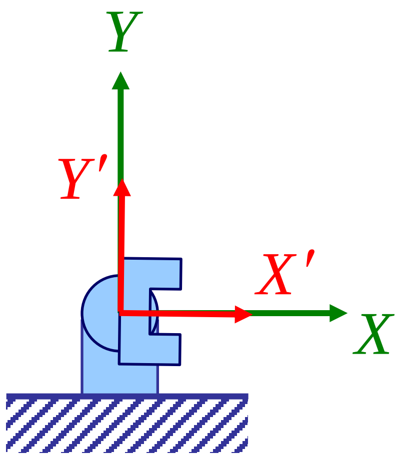
$$\begin{pmatrix} x_e \\ y_e \\ 1 \end{pmatrix} = \begin{pmatrix} & & \\ & T & \\ & & \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$$

$$T = (rot\theta_1)(transl_1)(rot\theta_2)(transl_2)$$

$$= \begin{pmatrix} \cos\theta_1 & -\sin\theta_1 & 0 \\ \sin\theta_1 & \cos\theta_1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & l_1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \cos\theta_2 & -\sin\theta_2 & 0 \\ \sin\theta_2 & \cos\theta_2 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & l_2 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Thinking of Transformations

- In a view of body-attached coordinate system

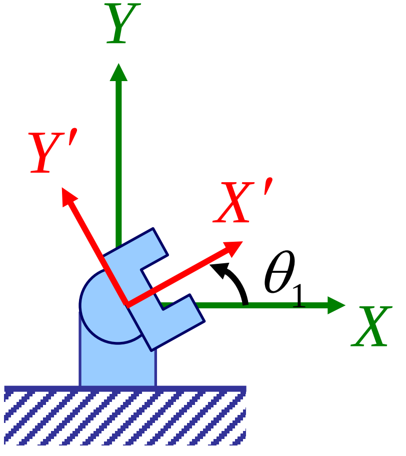


$$T = (rot\theta_1)(transl_1)(rot\theta_2)(transl_2)$$

$$= \begin{pmatrix} \cos\theta_1 & -\sin\theta_1 & 0 \\ \sin\theta_1 & \cos\theta_1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & l_1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \cos\theta_2 & -\sin\theta_2 & 0 \\ \sin\theta_2 & \cos\theta_2 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & l_2 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Thinking of Transformations

- In a view of body-attached coordinate system

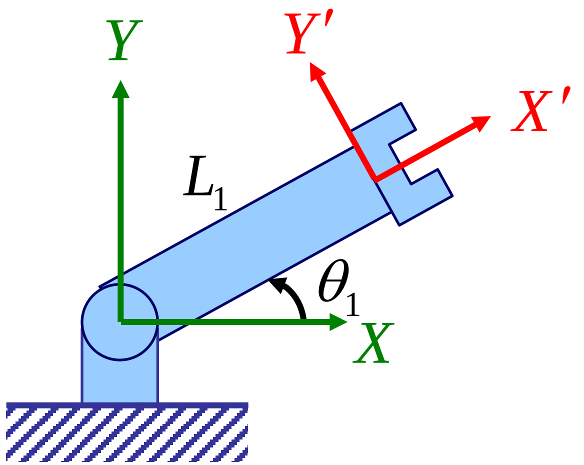


$$T = (rot\theta_1)(transl_1)(rot\theta_2)(transl_2)$$

$$= \begin{pmatrix} \cos\theta_1 & -\sin\theta_1 & 0 \\ \sin\theta_1 & \cos\theta_1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & l_1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \cos\theta_2 & -\sin\theta_2 & 0 \\ \sin\theta_2 & \cos\theta_2 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & l_2 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Thinking of Transformations

- In a view of body-attached coordinate system

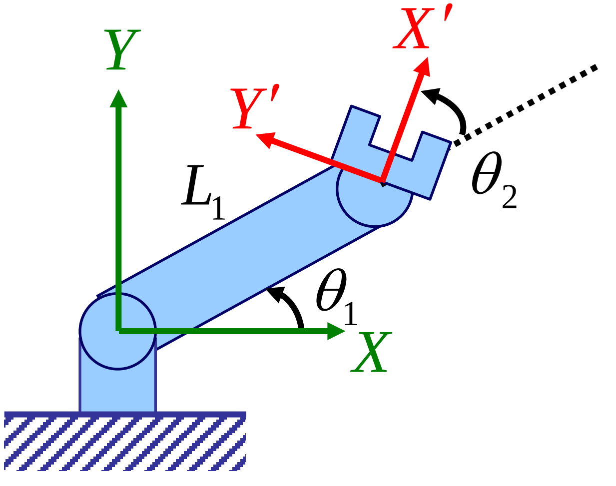


$$T = (rot\theta_1)(transl_1)(rot\theta_2)(transl_2)$$

$$= \begin{pmatrix} \cos\theta_1 & -\sin\theta_1 & 0 \\ \sin\theta_1 & \cos\theta_1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & l_1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \cos\theta_2 & -\sin\theta_2 & 0 \\ \sin\theta_2 & \cos\theta_2 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & l_2 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Thinking of Transformations

- In a view of body-attached coordinate system

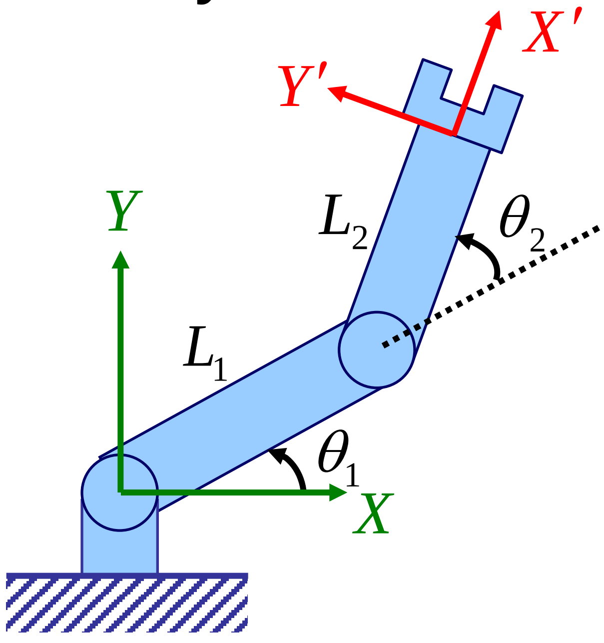


$$T = (rot\theta_1)(transl_1)(rot\theta_2)(transl_2)$$

$$= \begin{pmatrix} \cos\theta_1 & -\sin\theta_1 & 0 \\ \sin\theta_1 & \cos\theta_1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & l_1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \cos\theta_2 & -\sin\theta_2 & 0 \\ \sin\theta_2 & \cos\theta_2 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & l_2 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Thinking of Transformations

- In a view of body-attached coordinate system

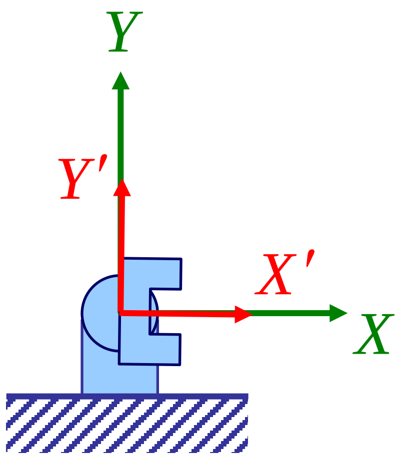


$$T = (rot\theta_1)(transl_1)(rot\theta_2)(transl_2)$$

$$= \begin{pmatrix} \cos\theta_1 & -\sin\theta_1 & 0 \\ \sin\theta_1 & \cos\theta_1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & l_1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \cos\theta_2 & -\sin\theta_2 & 0 \\ \sin\theta_2 & \cos\theta_2 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & l_2 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Thinking of Transformations

- In a view of global coordinate system

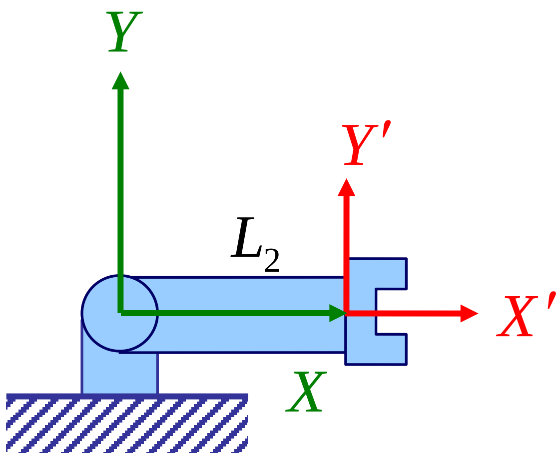


$$T = (rot\,\theta_1)(transl\,l_1)(rot\,\theta_2)(transl\,l_2)$$

$$= \begin{pmatrix} \cos\theta_1 & -\sin\theta_1 & 0 \\ \sin\theta_1 & \cos\theta_1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & l_1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \cos\theta_2 & -\sin\theta_2 & 0 \\ \sin\theta_2 & \cos\theta_2 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & l_2 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Thinking of Transformations

- In a view of global coordinate system

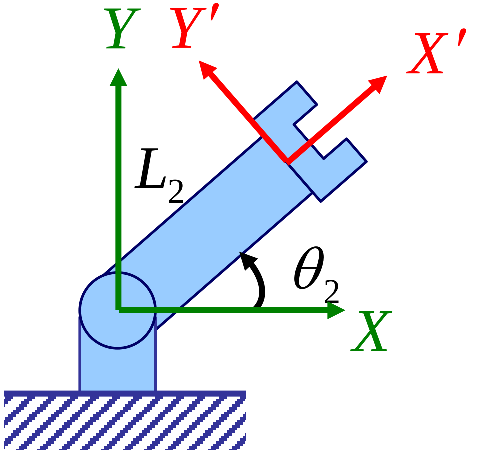


$$T = (rot\theta_1)(transl_1)(rot\theta_2)(transl_2)$$

$$= \begin{pmatrix} \cos\theta_1 & -\sin\theta_1 & 0 \\ \sin\theta_1 & \cos\theta_1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & l_1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \cos\theta_2 & -\sin\theta_2 & 0 \\ \sin\theta_2 & \cos\theta_2 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & l_2 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Thinking of Transformations

- In a view of global coordinate system

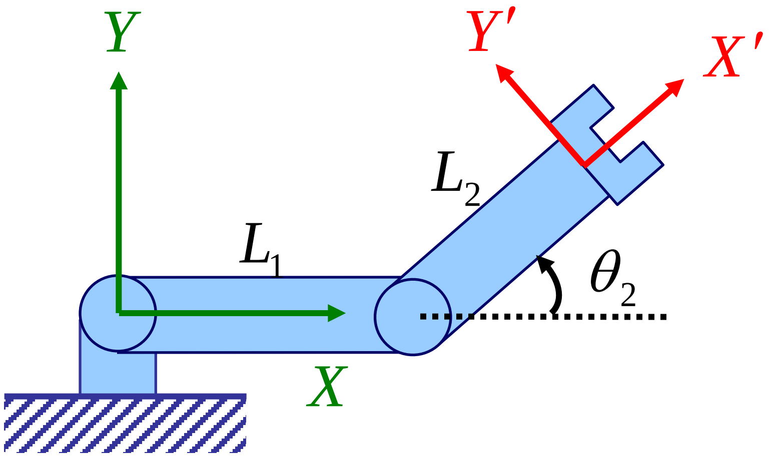


$$T = (rot\theta_1)(transl_1)(rot\theta_2)(transl_2)$$

$$= \begin{pmatrix} \cos\theta_1 & -\sin\theta_1 & 0 \\ \sin\theta_1 & \cos\theta_1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & l_1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \cos\theta_2 & -\sin\theta_2 & 0 \\ \sin\theta_2 & \cos\theta_2 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & l_2 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Thinking of Transformations

- In a view of global coordinate system

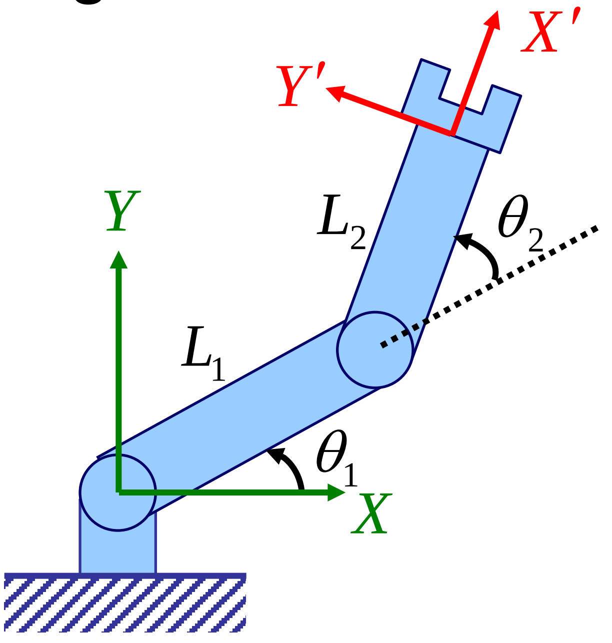


$$T = (rot\theta_1)(transl_1)(rot\theta_2)(transl_2)$$

$$= \begin{pmatrix} \cos\theta_1 & -\sin\theta_1 & 0 \\ \sin\theta_1 & \cos\theta_1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & l_1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \cos\theta_2 & -\sin\theta_2 & 0 \\ \sin\theta_2 & \cos\theta_2 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & l_2 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Thinking of Transformations

- In a view of global coordinate system



$$T = (rot\theta_1)(transl_1)(rot\theta_2)(transl_2)$$

$$= \begin{pmatrix} \cos\theta_1 & -\sin\theta_1 & 0 \\ \sin\theta_1 & \cos\theta_1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & l_1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \cos\theta_2 & -\sin\theta_2 & 0 \\ \sin\theta_2 & \cos\theta_2 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & l_2 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

How to Handle Ball-and-Socket Joints ?

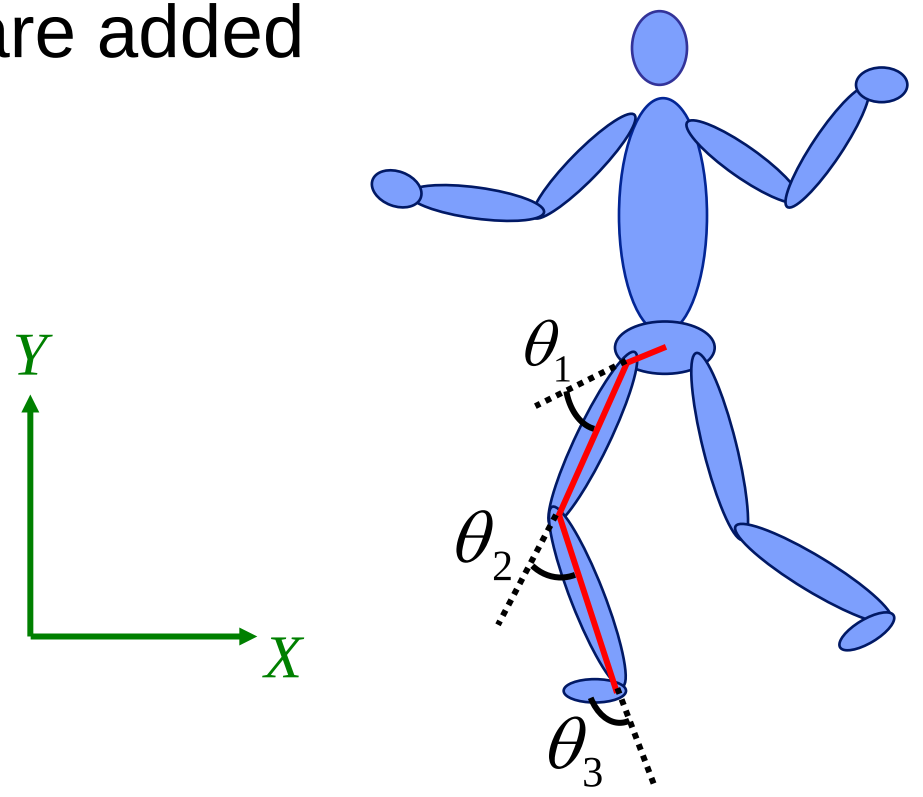
- Three revolute joints whose axes intersect at a point (equivalent to Euler angles), or
- 3D rotation about an arbitrary axis

$$T = (transl_1)(rot\theta_x)(rot\theta_y)(rot\theta_z)(transl_2)$$

$$= \cdots \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta_x & -\sin\theta_x & 0 \\ 0 & \sin\theta_x & \cos\theta_x & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \cos\theta_y & 0 & \sin\theta_y & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\theta_y & 0 & \cos\theta_y & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \cos\theta_z & -\sin\theta_z & 0 & 0 \\ \sin\theta_z & \cos\theta_z & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdots$$

Floating Base

- The position and orientation of the root segment are added



$$T = (rot\theta_r)(transl_r)(transl_0)(rot\theta_0)(transl_1)(rot\theta_1)(transl_2)(rot\theta_2)$$

Joint & Link Transformations

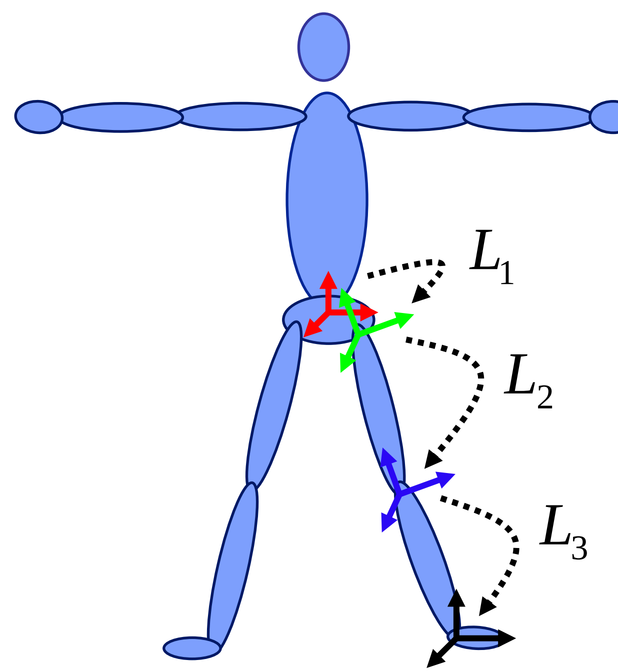
- Each segment has its own coordinate frame
- Forward kinematics map is an alternating multiple of
  - ***Joint transformations*** : represents joint movement
  - ***Link transformations*** : defines a frame relative to its parent

1st link transformation

$$T = J_0 L_1 J_1 L_2 J_2 L_3 J_3$$

1st joint transformation

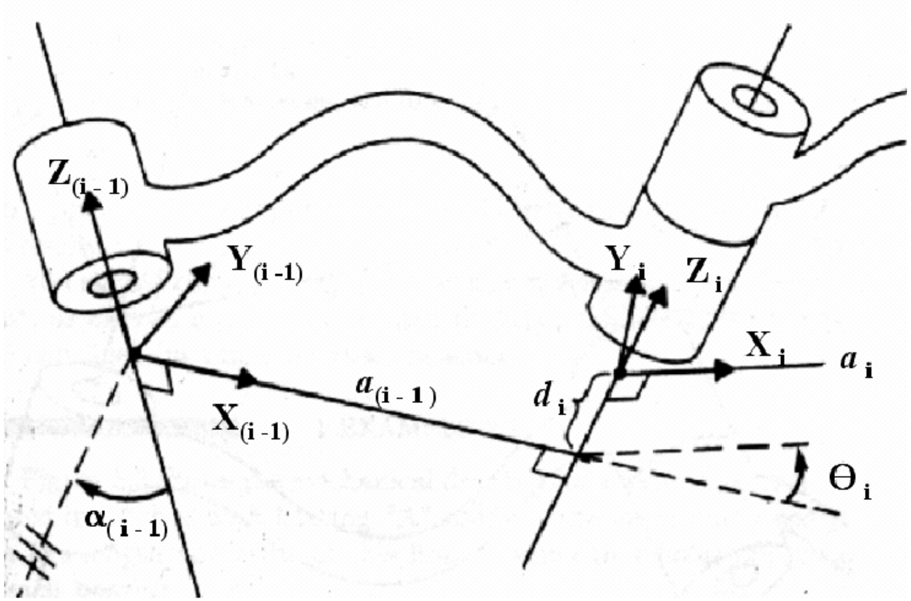
The position and orientation of the root segment



Joint & Link Transformations

- Both are rigid transformations in general
  - Joint transformations may include translation
    - Human joints are not ideal hinges
  - Link transformations may include rotation
    - Some links are twisted

Denavit-Hartenberg Notation



$$L_i = Rot(X, \alpha_{i-1}) \cdot Trans(X, a_{i-1}) \cdot Trans(Z, d_i) \cdot Rot(Z, \theta_i)$$

$a_i =$ the distance from Z_i to Z_{i+1} measured along X_i

$\alpha_i =$ the angle between Z_i and Z_{i+1} measured about X_i

$d_i =$ the distance from X_{i-1} to X_i measured along Z_i

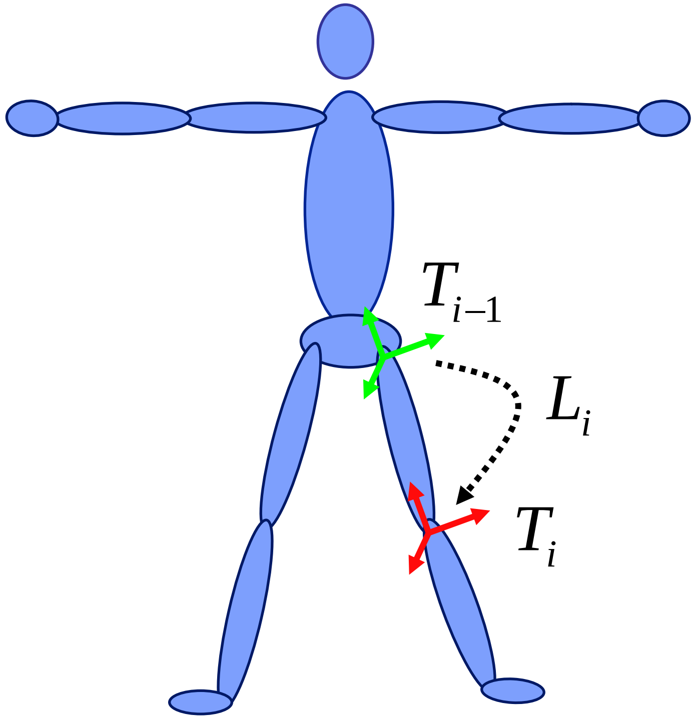
$\theta_i =$ the angle between X_{i-1} and X_i measured about Z_i

Link Transformations

- How do you compute the link transform for the i^{th} joint if you know the position and orientation of the i^{th} joint as well as its parent's position and orientation at the neutral pose ?

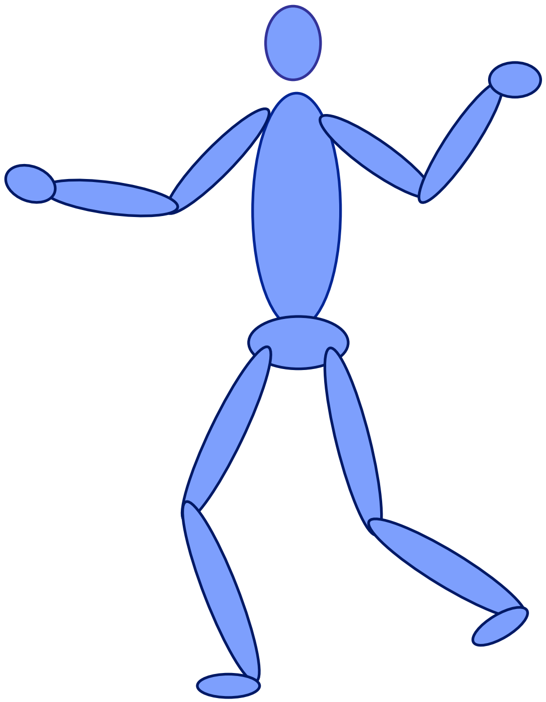
$$T_{i-1}L_i = T_i$$

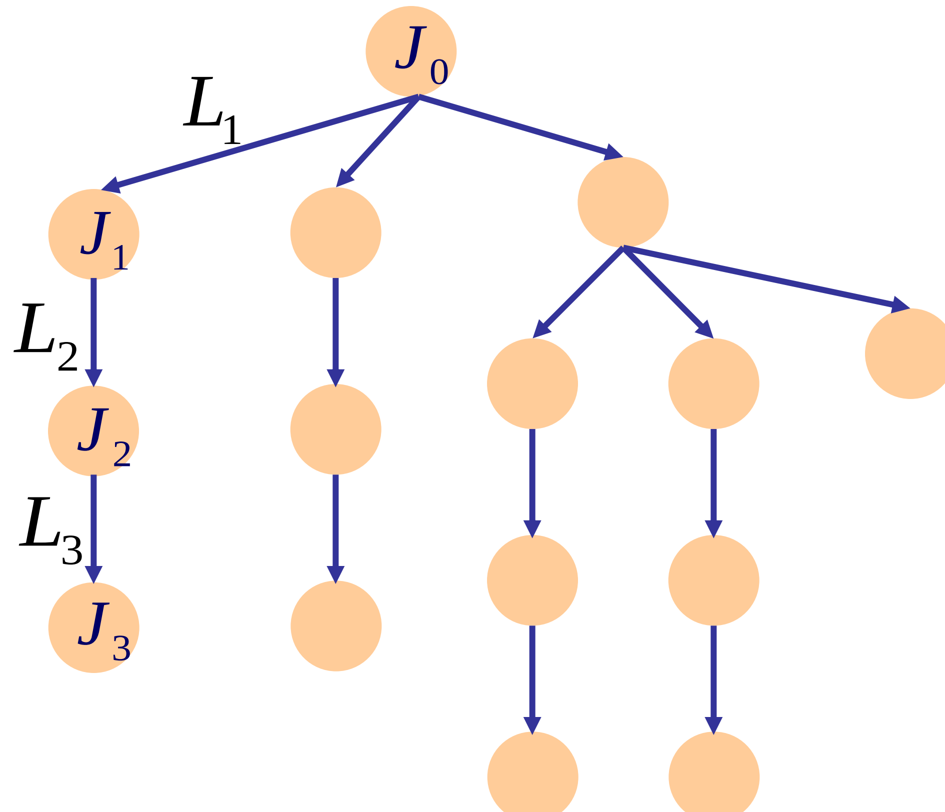
$$L_i = T_{i-1}^{-1}T_i$$



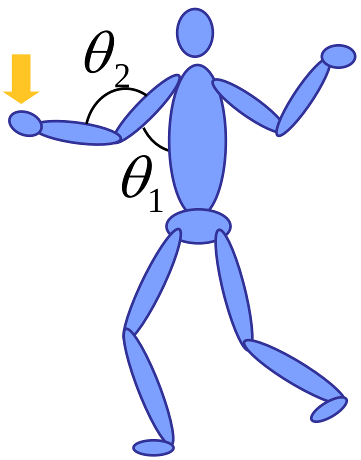
Representing Hierarchical Models

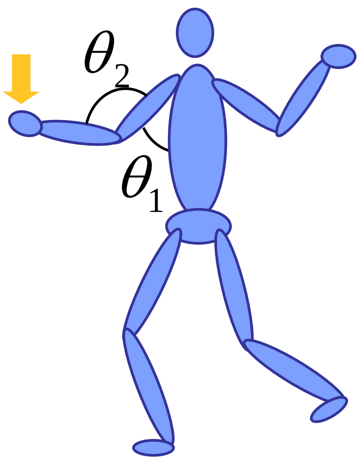
- A tree structure
  - A node contains a joint transformation
  - A arc contains a link transformation





Forward and Inverse Kinematics





$$(\mathbf{p}, \mathbf{q}) = \mathrm{F}(\theta_i)$$

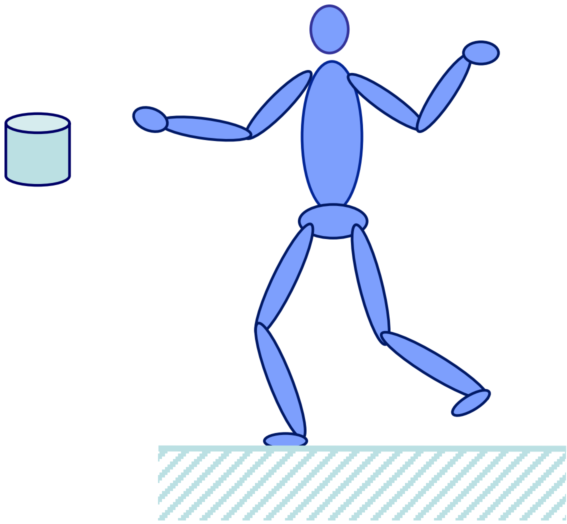
Forward Kinematics

$$\theta_i = \mathrm{F}^{-1}(\mathbf{p}, \mathbf{q})$$

Inverse Kinematics

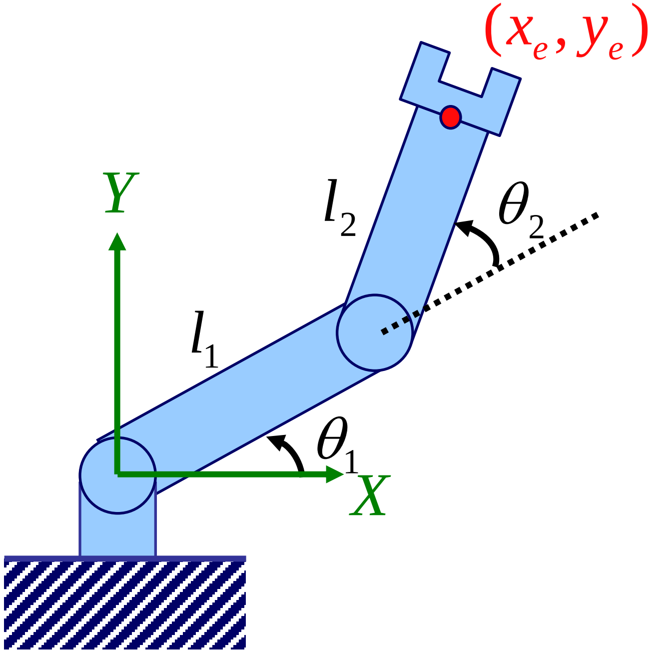
Why Inverse Kinematics ?

- Environmental interactions
  - Pick up an object or place feet on the ground
  - Hard to do with forward kinematics
- The pose of the character is described in the ***joint angle space***
- The environmental interaction is described in the ***work (Cartesian) space***

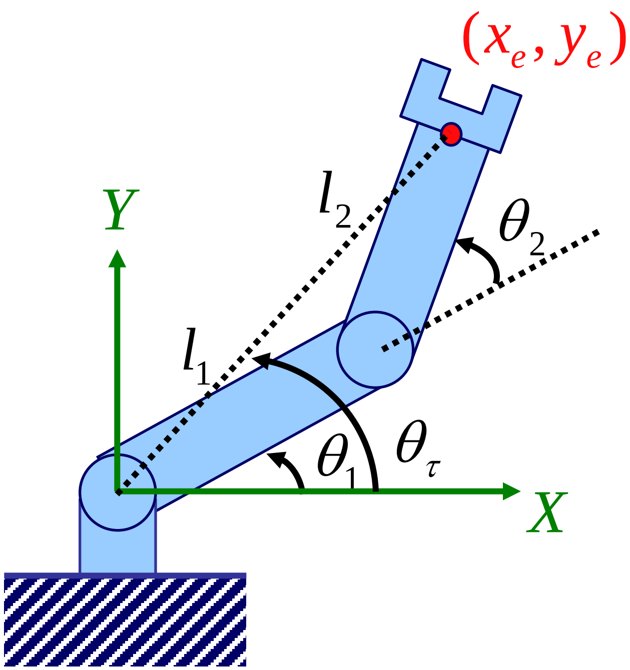


Inverse Kinematics: A Simple Example

- A simple robot arm in 2-dimensional space
  - Two revolute joints
  - The position of the end-effector is known
  - Compute joint angles



Analytic Solution for A Simple Example



$$\cos(\theta_\tau) = \frac{x_e}{\sqrt{x_e^2 + y_e^2}}$$

$$\theta_\tau = \cos^{-1}\left(\frac{x_e}{\sqrt{x_e^2 + y_e^2}}\right)$$

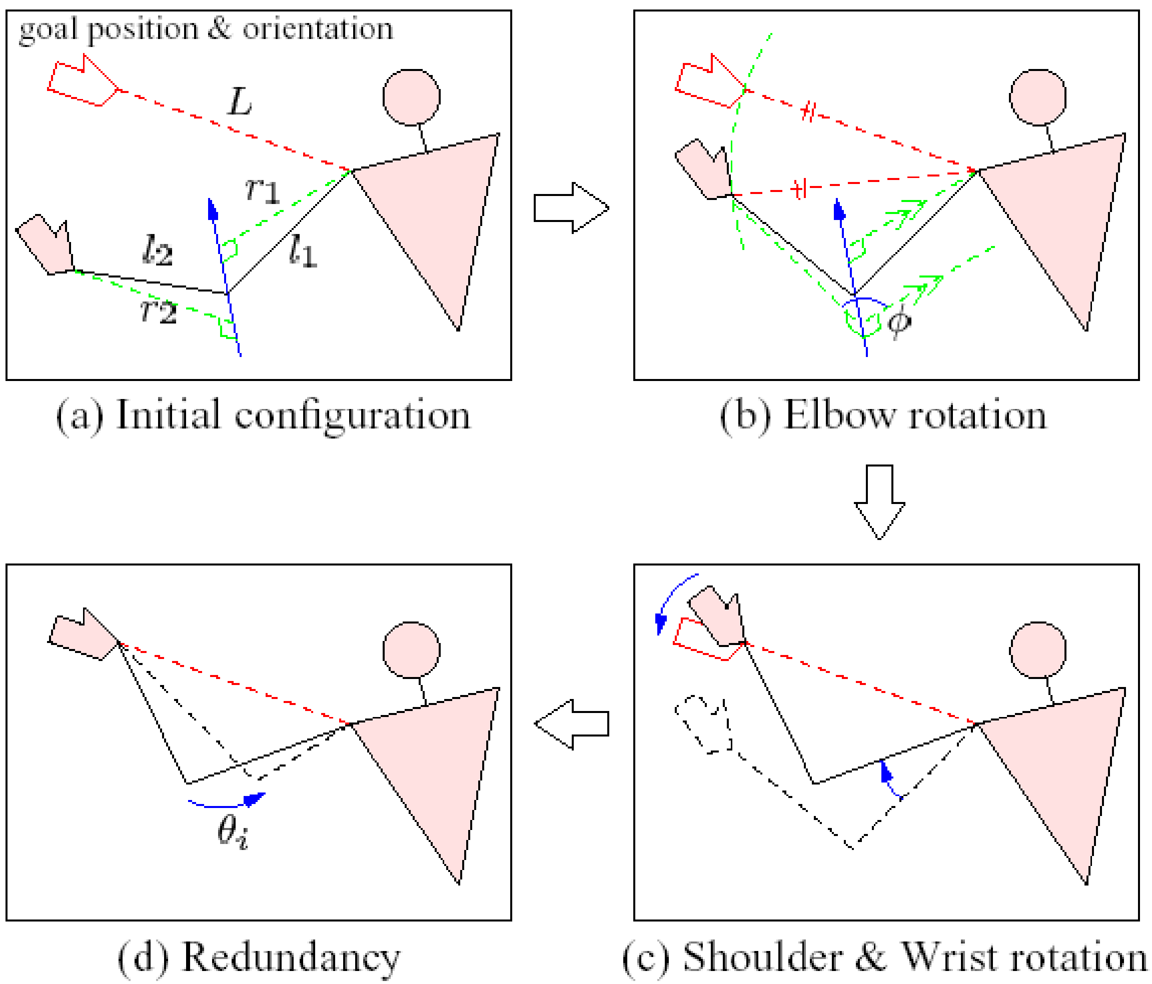
$$\cos(\theta_\tau - \theta_1) = \frac{l_1^2 + x_e^2 + y_e^2 - l_2^2}{2l_1\sqrt{x_e^2 + y_e^2}}$$

$$\theta_1 = \theta_\tau - \cos^{-1}\left(\frac{l_1^2 + x_e^2 + y_e^2 - l_2^2}{2l_1\sqrt{x_e^2 + y_e^2}}\right)$$

$$\cos(\pi - \theta_2) = \frac{l_1^2 + l_2^2 - x_e^2 - y_e^2}{2l_1 l_2}$$

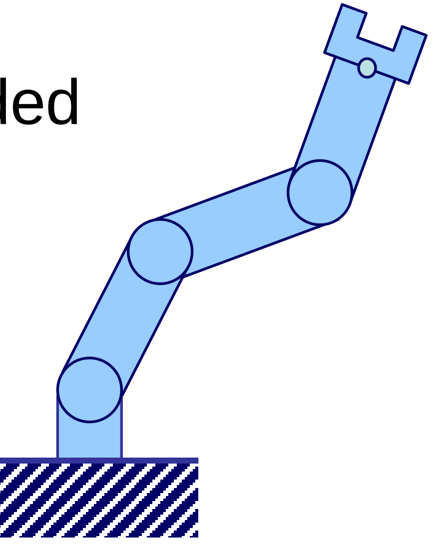
$$\theta_2 = \pi - \cos^{-1}\left(\frac{l_1^2 + l_2^2 - x_e^2 - y_e^2}{2l_1 l_2}\right)$$

Redundancy in Human Arms



Why so difficult to get a closed-form solution ?

- Redundancies
  - Multiple solutions (# of unknowns > # of equations)
- Joint limit
  - The range of each unknown is bounded
- Reachable workspace
  - No solution, or
  - A unique solution, or
  - Multiple solutions
- Multiple goals
  - Four limbs may have constraints simultaneously
  - Intermediate links can also be constrained

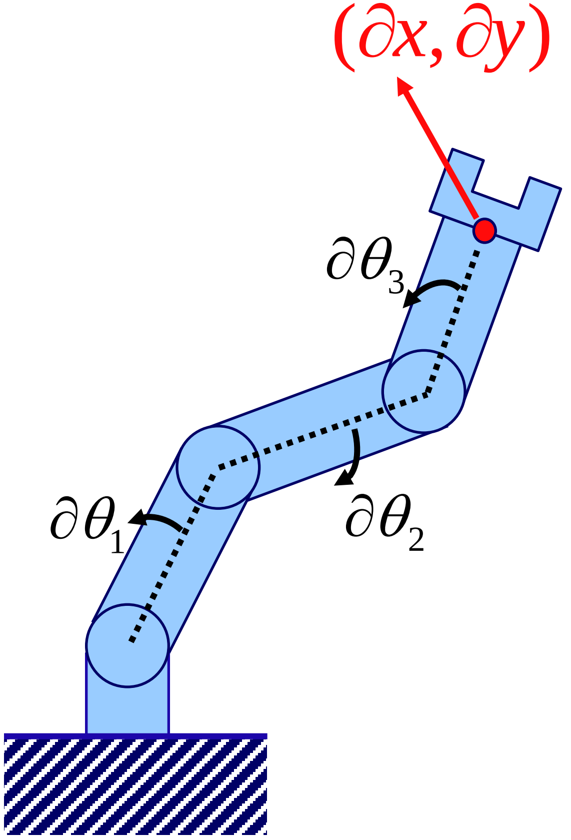


Iterative Methods

- Iteratively refine joint angles to the goal
  - Consider infinitesimal changes

$$\text{compute } (\theta_1, \theta_2, \theta_3) \text{ from } (x, y)$$

$$\text{compute } (\partial\theta_1, \partial\theta_2, \partial\theta_3) \text{ from } (\partial x, \partial y)$$



Jacobians of Forward Kinematics Map

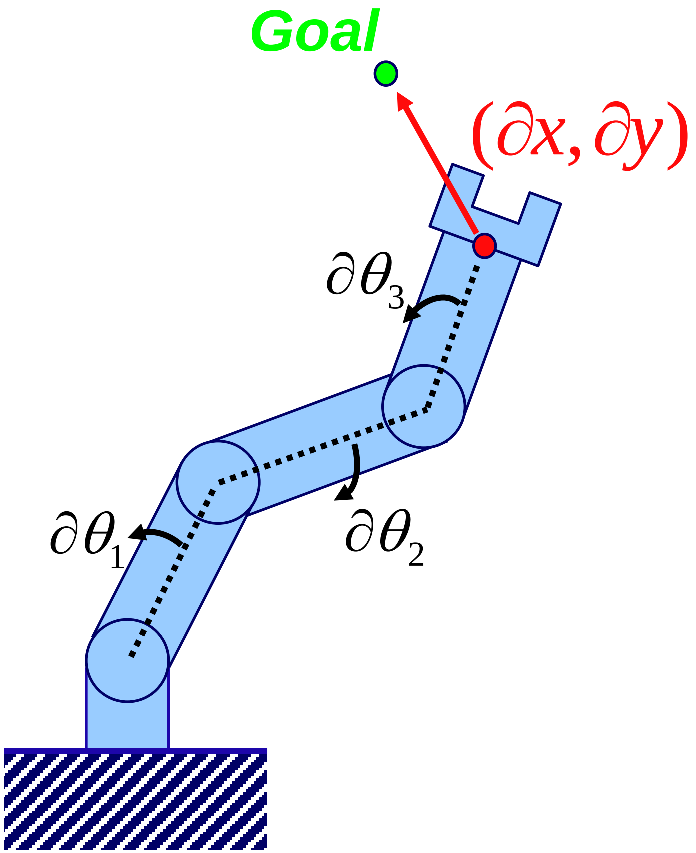
- Forward Kinematics Map

$$(x, y) = (F_x, F_y) = F(\theta_1, \theta_2, \theta_3)$$

- Jacobian

$$\begin{pmatrix} \dot{x} \\ \dot{y} \end{pmatrix} = \mathbf{J} \begin{pmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \end{pmatrix}$$

$$= \begin{pmatrix} \frac{\partial F_x}{\partial \theta_1} & \frac{\partial F_x}{\partial \theta_2} & \frac{\partial F_x}{\partial \theta_3} \\ \frac{\partial F_y}{\partial \theta_1} & \frac{\partial F_y}{\partial \theta_2} & \frac{\partial F_y}{\partial \theta_3} \end{pmatrix} \begin{pmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \end{pmatrix}$$

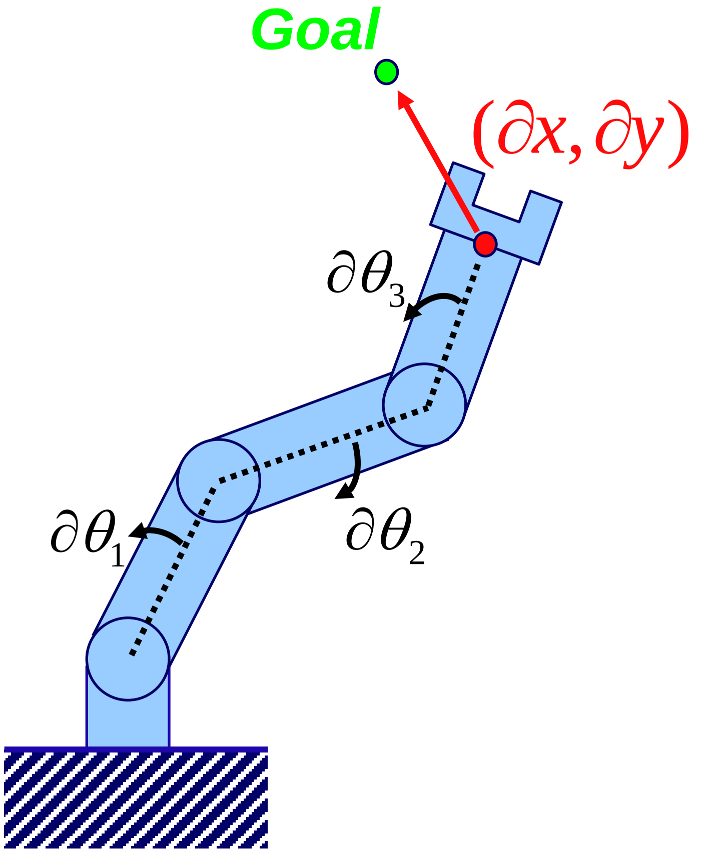


Jacobian of Forward Kinematics Map

- If the inverse of the Jacobian can be computed, …

$$\begin{pmatrix}\theta_1\\ \theta_2\\ \theta_3\end{pmatrix}_{t+1} = \begin{pmatrix}\theta_1\\ \theta_2\\ \theta_3\end{pmatrix}_t + \Delta t \begin{pmatrix}\dot{\theta}_1\\ \dot{\theta}_2\\ \dot{\theta}_3\end{pmatrix}_t$$

$$= \begin{pmatrix}\theta_1\\ \theta_2\\ \theta_3\end{pmatrix}_t + \Delta t \mathbf{J}^{-1} \begin{pmatrix}\dot{x}\\ \dot{y}\end{pmatrix}_t$$



A System of Linear Equations

$$\mathbf{J}\dot{\boldsymbol{\theta}} = \dot{\mathbf{x}}$$

- # of unkowns = the dimension of $\boldsymbol{\theta}$ = n
- # of equations = the dimension of $\boldsymbol{x}$ = m
- $\boldsymbol{J}$ is a (m x n) matrix

- The solution maybe unique if $m=n$
- The linear system is under-specified if $m<n$
- The linear system is over-specified if $m>n$

the pseudoinverse of Jacobian is required

Redundancy

- A single solution must be chosen from multiple solutions
  - **"Closest" to the current configuration**
    - Pseudo inverse minimizes joint angle rates (locally)
  - **Move outermost links the most**
    - The outermost link sweeps a smallest region (visual change)
  - **Minimum time and effort**
    - Dynamics involves
  - **Secondary goal**
    - Additional constraints
  - **Natural looking**
    - Biomechanical experiments

Summary

- Very simple structure allows an analytic solution
- Most of complex articulated figures requires a numerical solution
- May not always get the “right” answer
  - Need to tweak the solution later