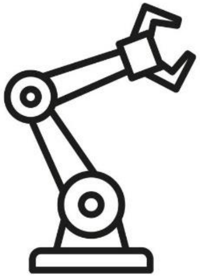
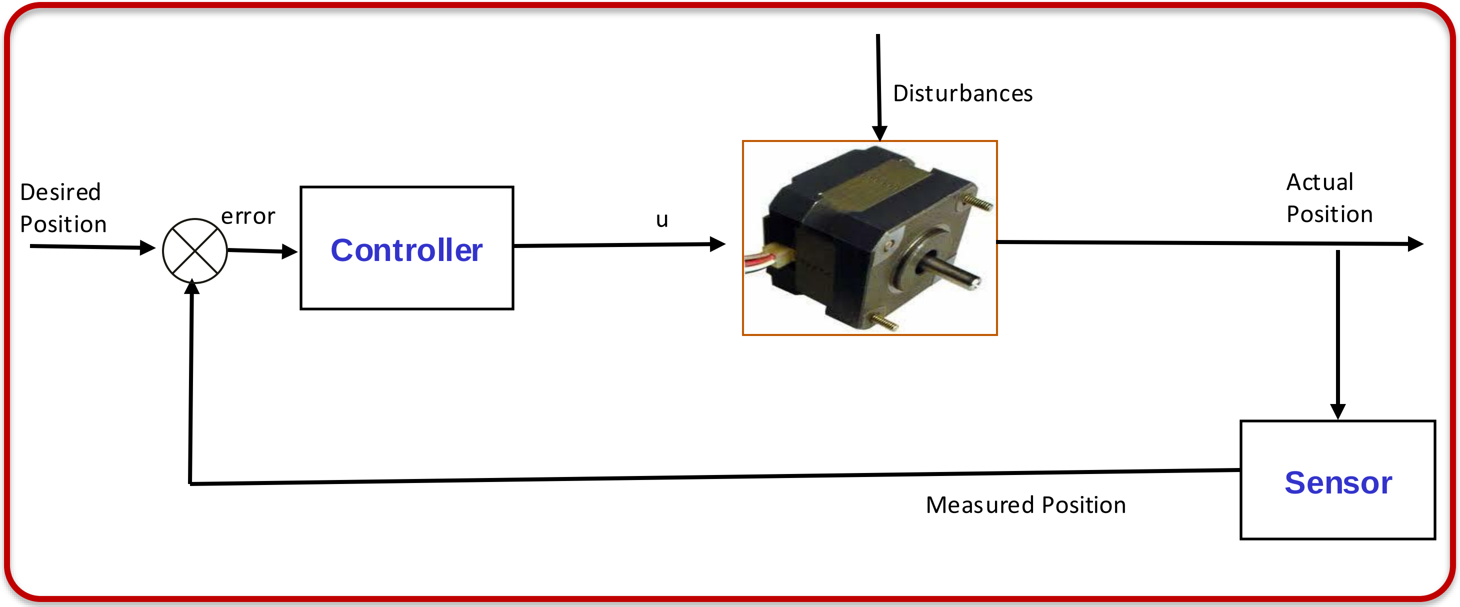
RBT350
GATEWAY TO ROBOTICS

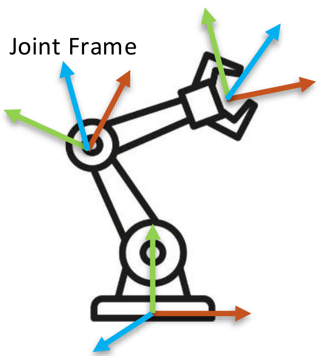
Frames and Poses

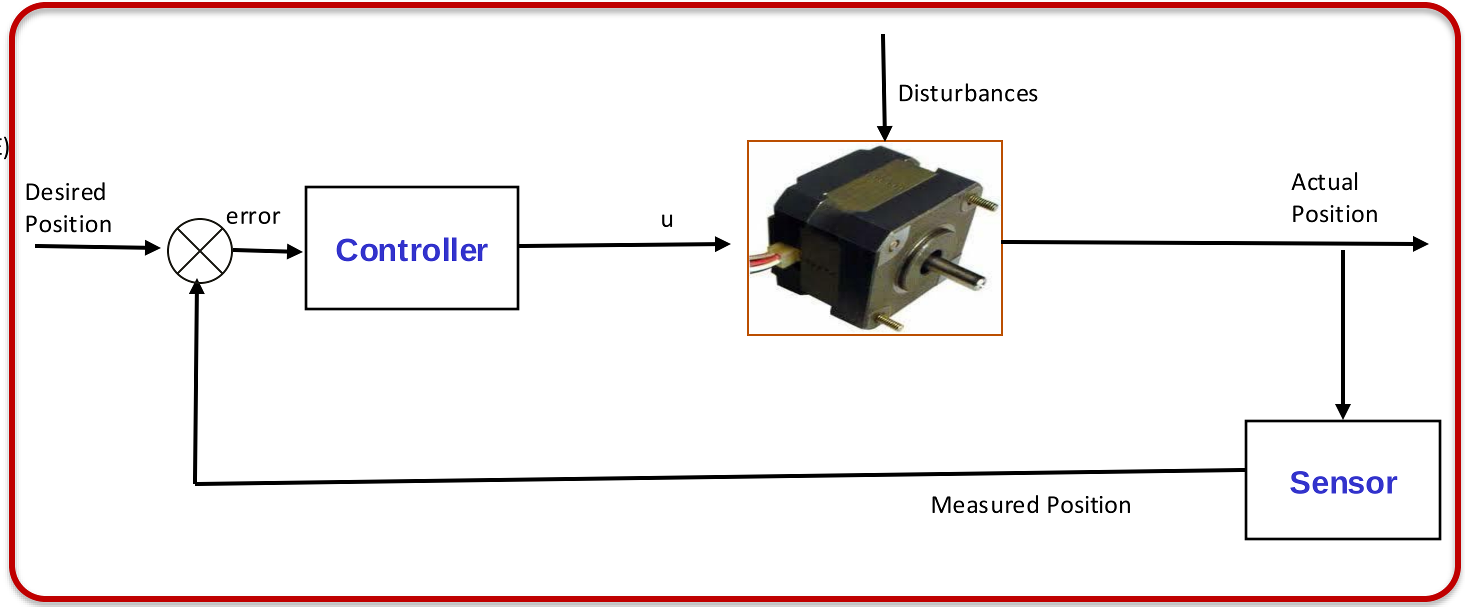
So now we can control the joints to move…





…but move where? What is the goal?



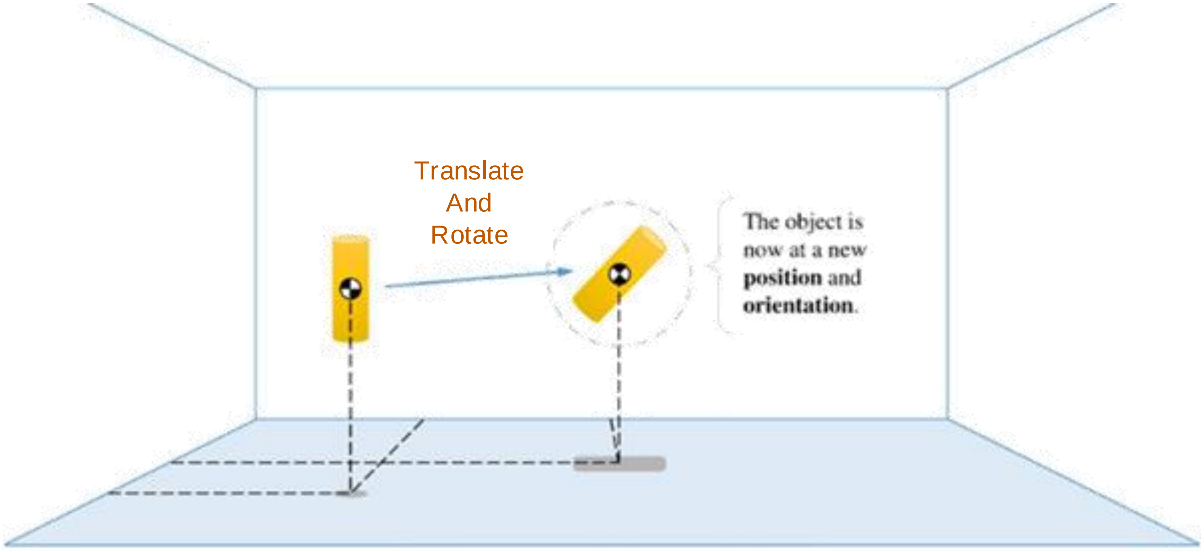


Tasks are usually defined in 3D Cartesian Space not joint space
We want to move a frame on the robot to a desired pose
We need to deal with poses and motion in Cartesian space

What will you learn today?

- Preliminaries
  - How to represent the state of an object in Cartesian space? → Poses
  - We will learn later how to connect Joint Space and Cartesian space
- How to define a frame's orientation and pose
  - Translations
  - Rotation Matrices
  - Exponential Coordinates
  - Euler Angles (Intrinsic vs Extrinsic Rotations)
  - Quaternions

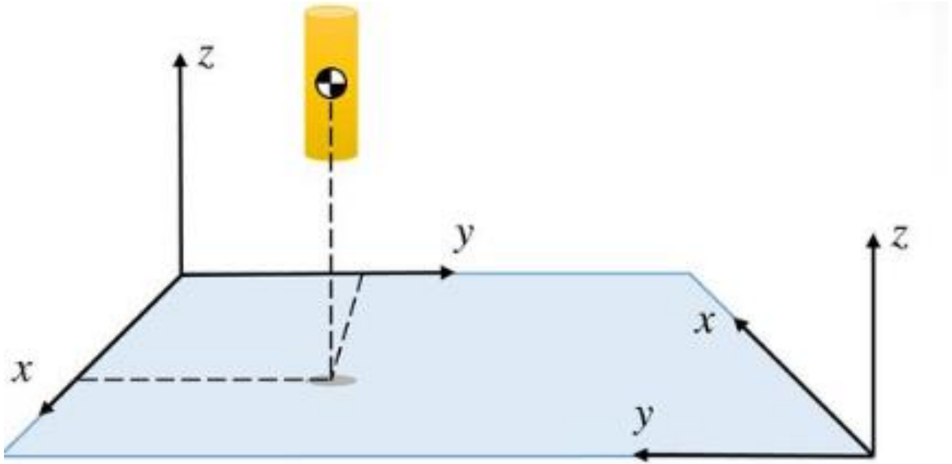
Define: Translation (Move) and Rotation (Rotate)

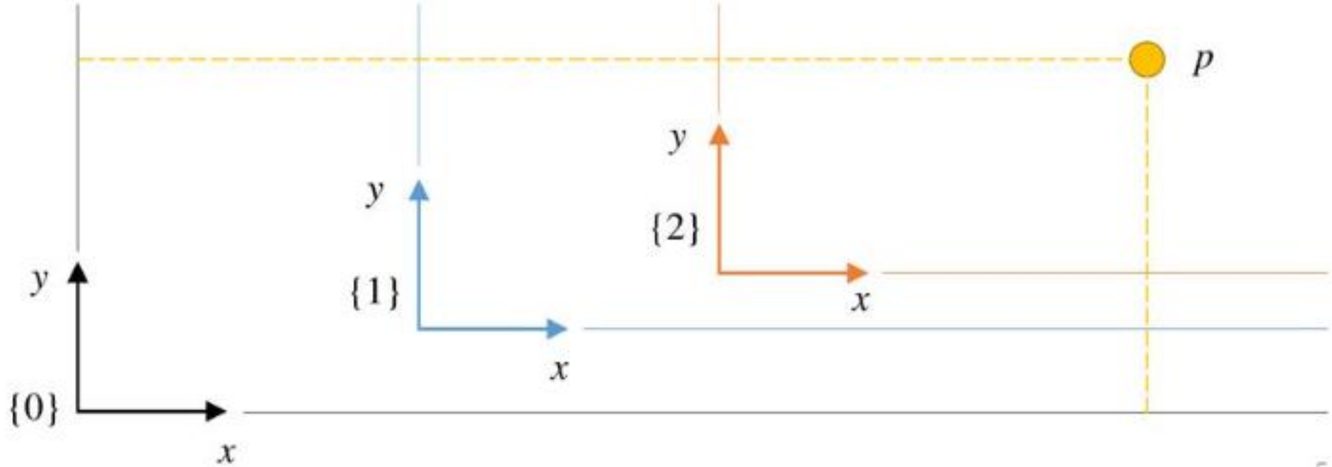


How do we express how much we move or rotate?
With respect to what?

Define: Reference Frames

- A point ***p*** can have different coordinates in different frames
- Frames can be fixed to objects of attached to moving objects



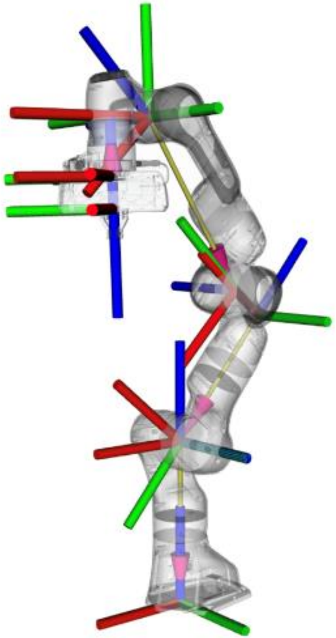


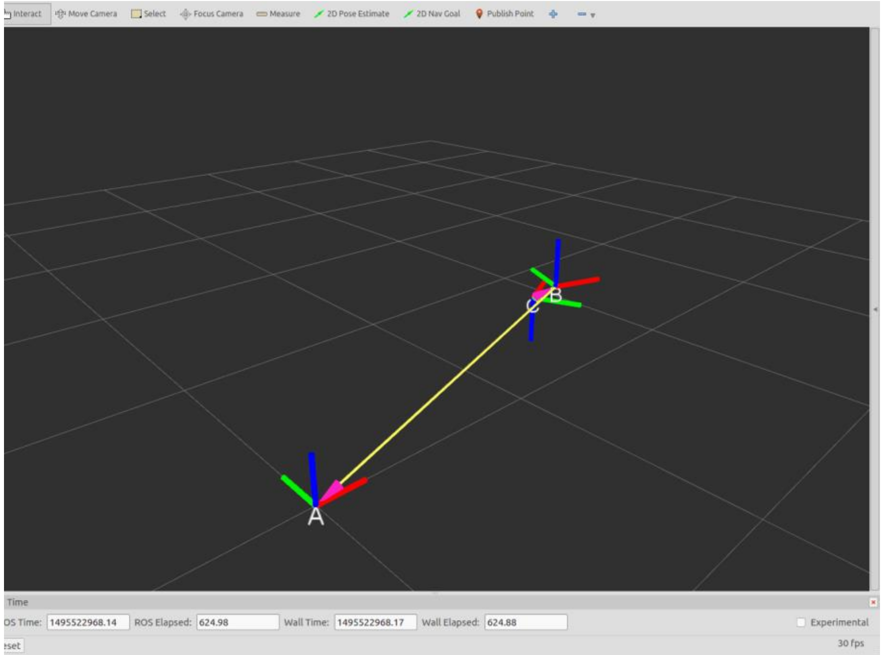
$$\mathbf{p} = \vec{p} = \begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix} \in \mathbb{R}^3$$

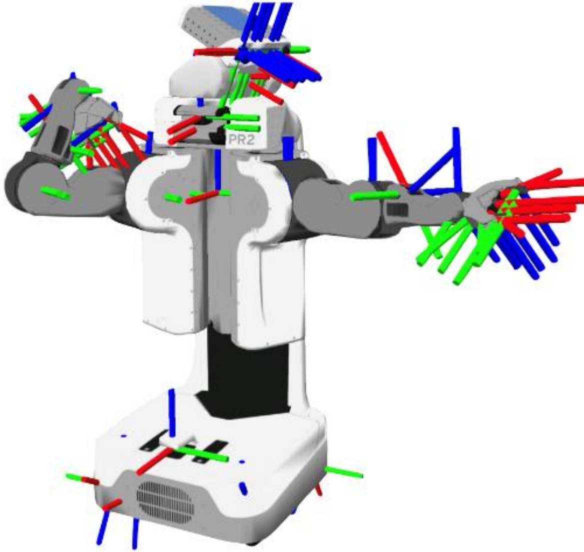
$${}^0\mathbf{p} \neq {}^1\mathbf{p} \neq {}^2\mathbf{p}$$

But it is the same point.

Translations and Rotations in Robotics… Many Frames







Preliminaries

- Define: Vector: element of $\mathbb{R}^n$

$$\mathbf{p} = \vec{p} = \begin{bmatrix} p_1 \\ p_2 \\ \vdots \\ p_n \end{bmatrix} \in \mathbb{R}^n$$

- Define: Inner product

$$\langle \mathbf{p}\mathbf{q} \rangle = \mathbf{p}\cdot\mathbf{q} = \sum_{i=0}^{n-1} p_i q_i = \|\mathbf{p}\|_2 \|\mathbf{q}\|_2 \cos(\theta)$$

- Define: Vector 2-norm (L2)

$$\|\mathbf{p}\|_2 = \left[p_1^2 + p_2^2 + \cdots + p_n^2 \right]^{1/2}$$

- Define: angle between two vectors

$$\cos(\theta) = \frac{\mathbf{p}\cdot\mathbf{q}}{\|\mathbf{p}\|_2 \|\mathbf{q}\|_2}$$

Preliminaries

- Define: Unit Vectors

$$\hat{\mathbf{i}} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \in \mathbb{R}^3$$

$$\hat{\mathbf{j}} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \in \mathbb{R}^3$$

$$\hat{\mathbf{k}} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \in \mathbb{R}^3$$

- Common Notation Simplification

$$s_\theta = \sin(\theta)$$
$$c_\theta = \cos(\theta)$$
$$s_{12} = \sin(\theta_1 + \theta_2)$$
$$c_{123} = \cos(\theta_1 + \theta_2 + \theta_3)$$
$$etc.$$

- Frame Designations (sources will vary)

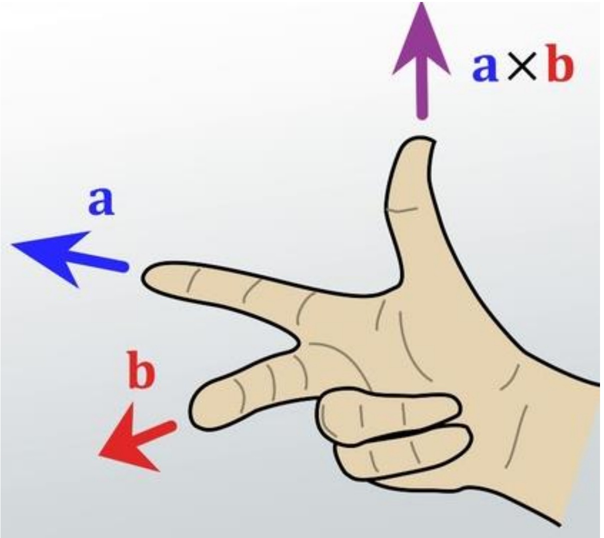
$$^{1}\mathbf{p} \quad {}_{1}\mathbf{p} \quad {}^{1}_{2}\mathbf{p}$$

Preliminaries

- Cross product (result is another vector!)

$$\|p \times q\| = \|p\|\|q\|\sin(\theta)$$

$$\mathbf{p} \times \mathbf{q} = \begin{bmatrix} p_2 q_3 - p_3 q_2 \\ p_3 q_1 - p_1 q_3 \\ p_1 q_2 - p_2 q_1 \end{bmatrix}$$



Preliminaries

Matrix Algebra:

- Identity matrix: $A = I \cdot A$
- Some matrices have an inverse, such that: $AA^{-1} = I$
- Inversion is tricky: $(ABC)^{-1} = C^{-1}B^{-1}A^{-1}$ (Derived from non-commutativity property)

$$I = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \in \mathbb{R}^{3x3}$$

- Basic Operations (for matrices in $\mathbb{R}^{2x2}$)

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} + \begin{bmatrix} e & f \\ g & h \end{bmatrix} = \begin{bmatrix} a+e & b+f \\ c+g & d+h \end{bmatrix}$$

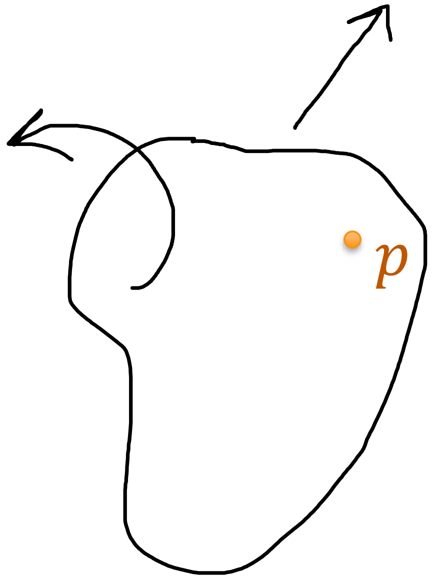
$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} - \begin{bmatrix} e & f \\ g & h \end{bmatrix} = \begin{bmatrix} a-e & b-f \\ c-g & d-h \end{bmatrix}$$

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} e & f \\ g & h \end{bmatrix} = \begin{bmatrix} ae+bg & af+bh \\ ce+dg & cf+dh \end{bmatrix}$$

Matrix Multiplication is NOT communicative!!! (but it is associative)

What is our goal?

- Given a rigid body we want to be able to express how it moves in a concise way
  - When it translates
  - When it rotates
- That means, if I know the coordinates of one point on the rigid body before it moves, we will apply <u>*a transformation*</u> that will tell us what are the coordinates of the point after the motion
- Different types of motion have different representations for the transformation



Representations of Translations in 3D Space

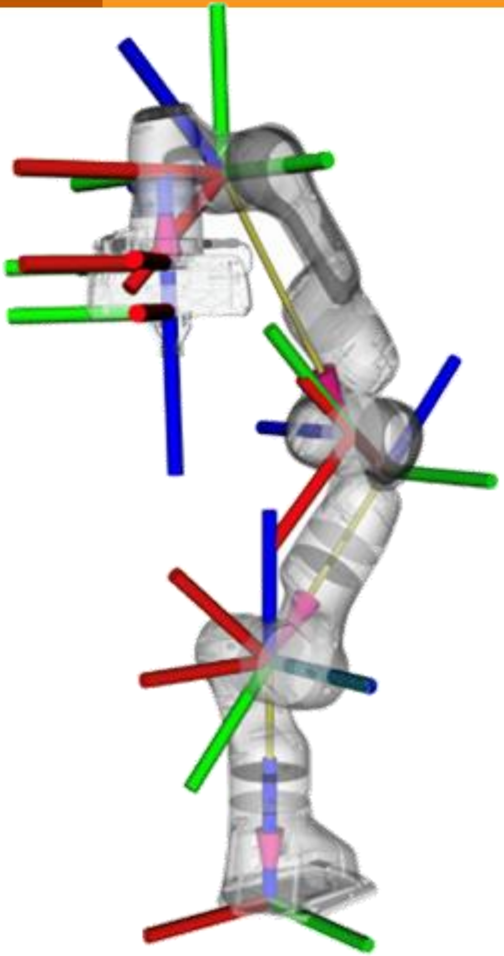
1. Displacement Vector → $\mathbb{R}^3$ Group

Representations of Rotations in 3D Space

1. Axis-Angle
2. Euler Angles
3. Rotation Matrix
4. Quaternions

Representations of Translation+Rotation in 3D Space

1. Transformation (Homogeneous) Matrices → SE(3) Group
2. Twists → se(3) Algebra



Representations of Translations in 3D Space

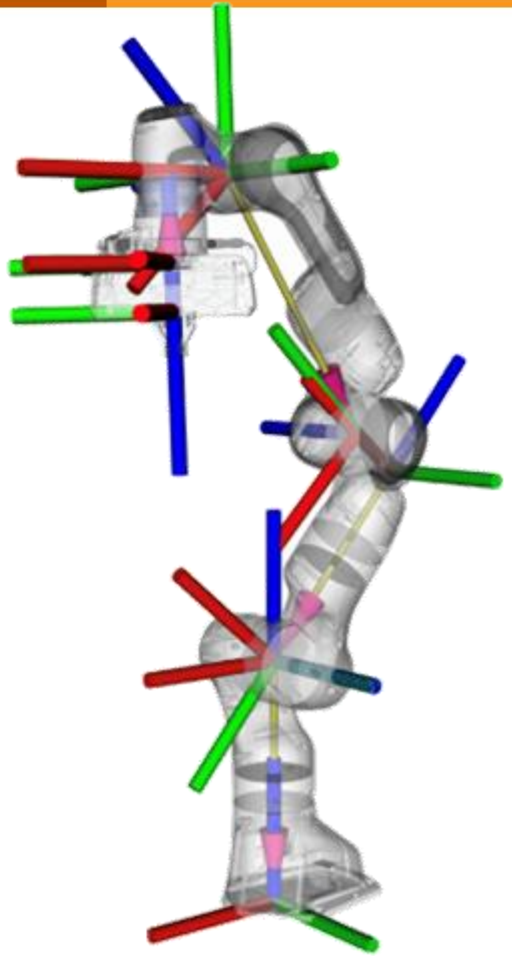
1. Displacement Vector → $\mathbb{R}^3$ Group

Representations of Rotations in 3D Space

1. Axis-Angle
2. Euler Angles
3. Rotation Matrix
4. Quaternions

Representations of Translation+Rotation in 3D Space

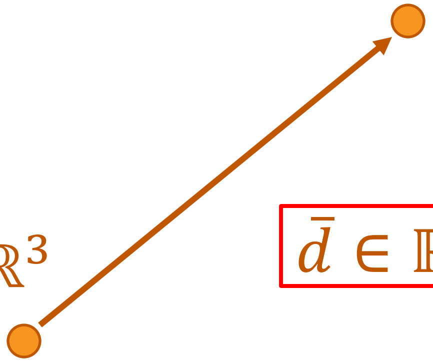
1. Transformation (Homogeneous) Matrices → SE(3) Group
2. Twists → se(3) Algebra



Translation

- Move a point to a new location

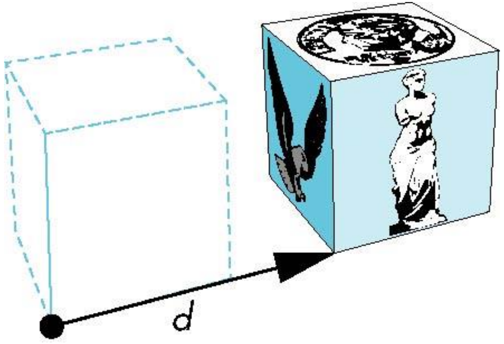
$$p' = p + \bar{d}$$



$p \in \mathbb{R}^3$

$\boxed{\bar{d} \in \mathbb{R}^3}$

- Move an object to a new location



Every point displaced by same vector

Translation → 3 Degrees of Freedom (DoF)

Representations of Translations in 3D Space

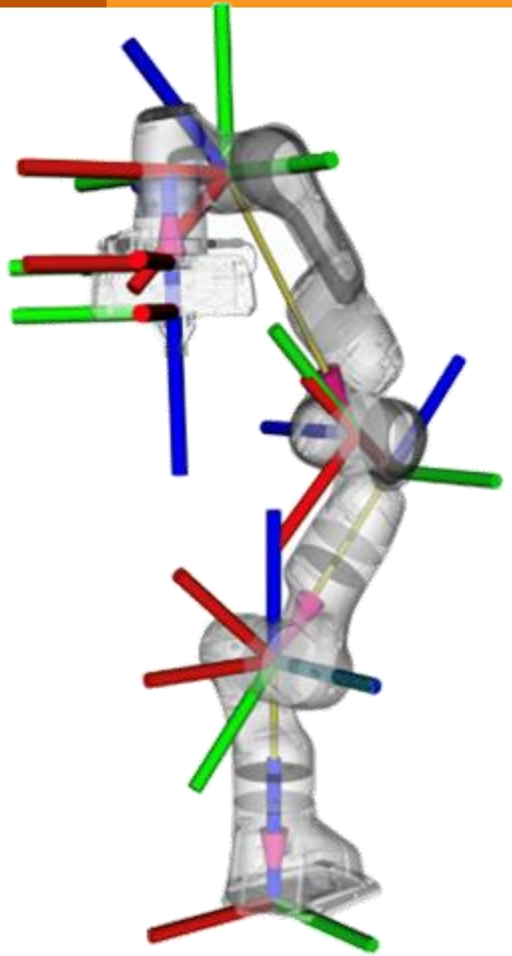
1. Displacement Vector → $\mathbb{R}^3$ Group

Representations of Rotations in 3D Space

1. Axis-Angle
2. Euler Angles
3. Rotation Matrix
4. Quaternions

Representations of Translation+Rotation in 3D Space

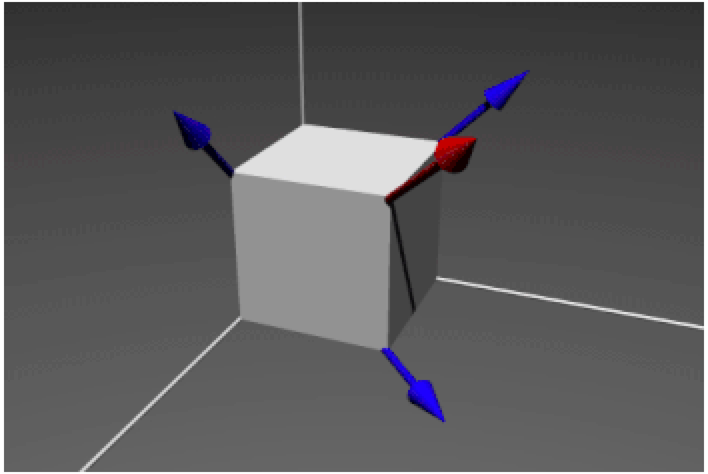
1. Transformation (Homogeneous) Matrices → SE(3) Group
2. Twists → se(3) Algebra

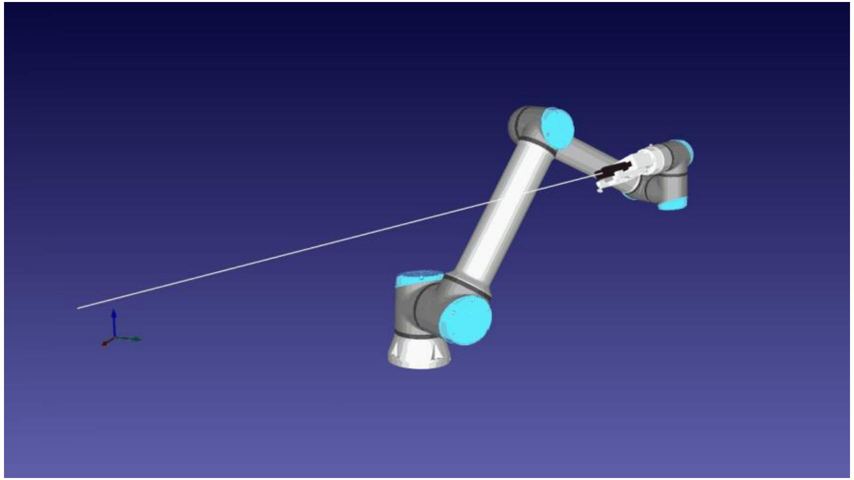


Rotation: defined

- The displacement of a rigid body that leaves *at least* on point fixed in space

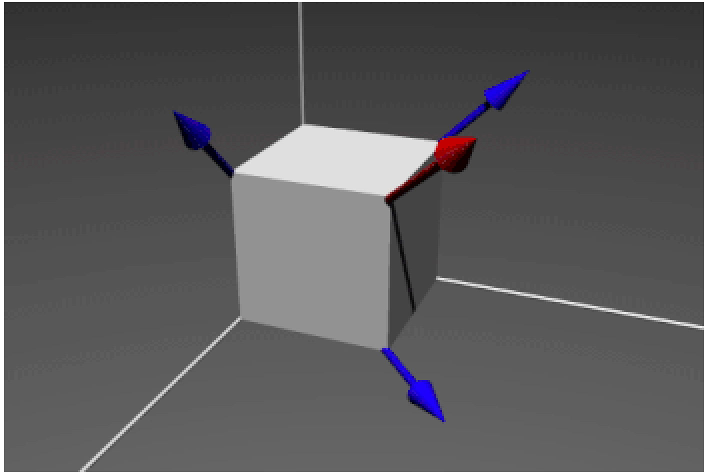
- Critical in robotics since commonly each joint rotates in a frame fixed to the previous link.

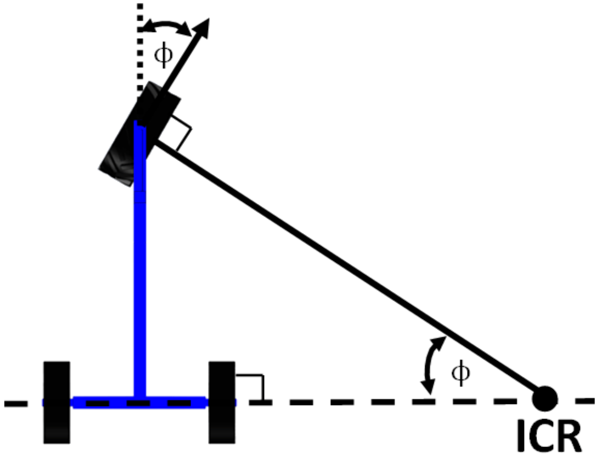




Rotation: defined

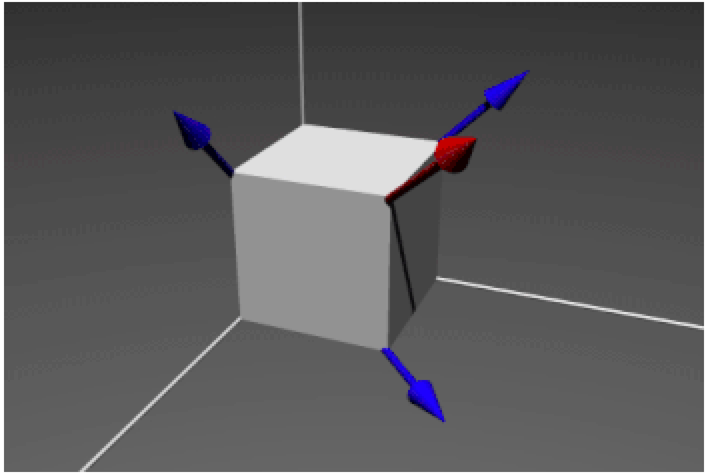
- The displacement of a rigid body that leaves *at least* on “point” fixed in space
- This point may not even be on the rigid body (see the car)
- The current rigid point is known as the “Instantaneous Center of Rotation”



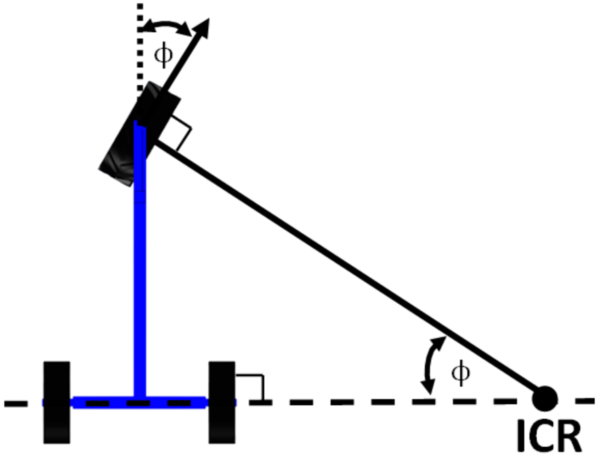


Rotation: defined

- In 3D spatial systems (E^3) a line is fixed
- That line is called the axis of rotation



- In planar systems (E^2), a point is fixed.
- The point is the center of rotation



Representations of Translations in 3D Space

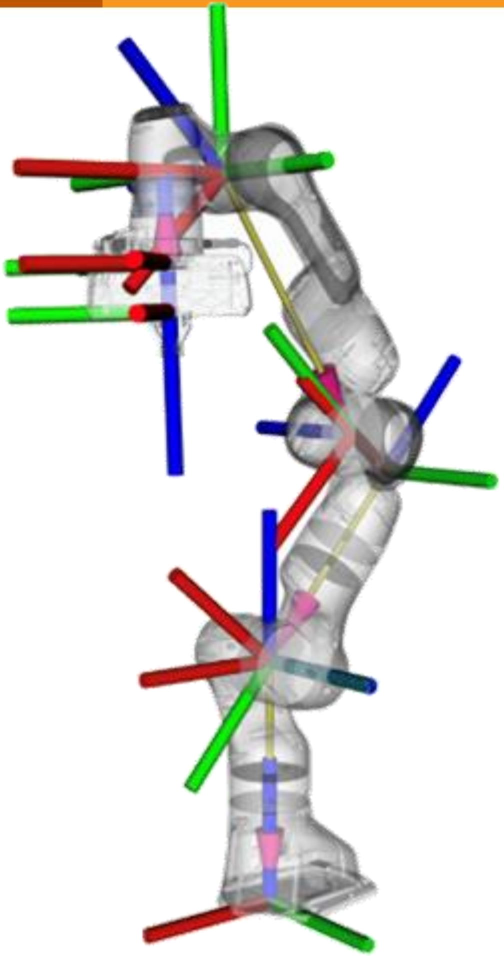
1. Displacement Vector → $\mathbb{R}^3$ Group

Representations of Rotations in 3D Space

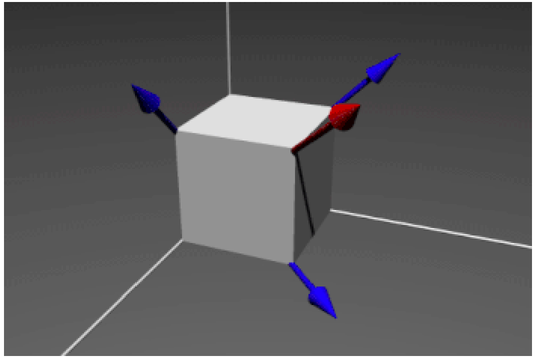
1. Axis-Angle
2. Euler Angles
3. Rotation Matrix
4. Quaternions

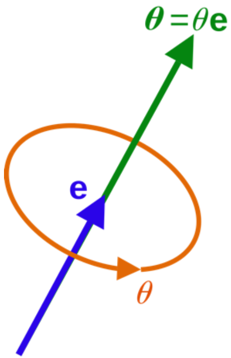
Representations of Translation+Rotation in 3D Space

1. Transformation (Homogeneous) Matrices → SE(3) Group
2. Twists → se(3) Algebra



In 3D Space, a Rotation happens around a Fixed Axis





- Euler Theorem: "Every spatial rotation leaves some line fixed."
- Representations:
  - A free unit vector $e \in \mathbb{R}^3$ indicates direction of that line.
  - A scalar θ is the magnitude of a clockwise rotation
- Good to integrate angular velocities:
  - If a rigid body rotates with constant angular velocity $\overline{\omega}$ [rad/s], after time t it will have rotated $\widehat{\omega} \cdot |\overline{\omega}| t$

Properties of Axis-Angle. They are…

Simple and intuitive. GOOD!

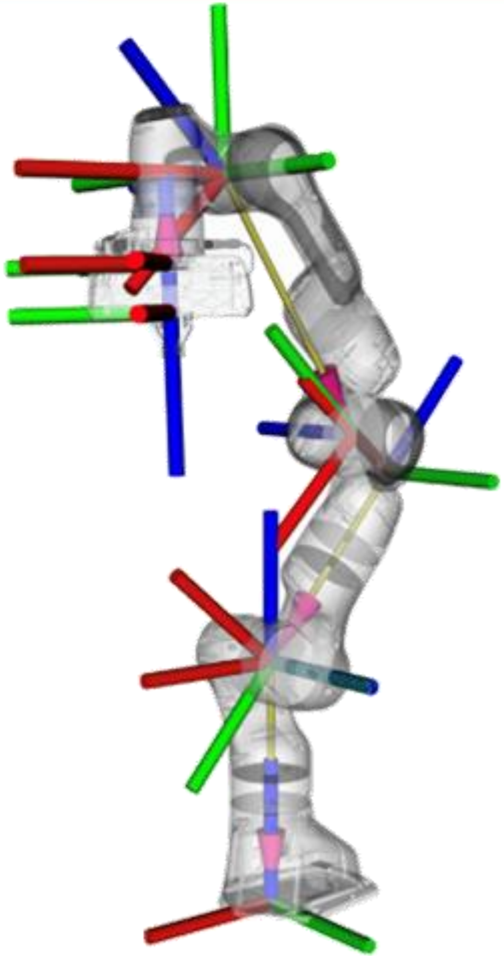
Easy to integrate (angular velocity -> rotation). GOOD!

Redundant and not minimal (4 values for 3 DoFs). BAD!

They do not interpolate in an obvious way. BAD!

Not easy to combine. BAD!

They cannot be directly applied to “move” points. BAD!



Representations of Translations in 3D Space

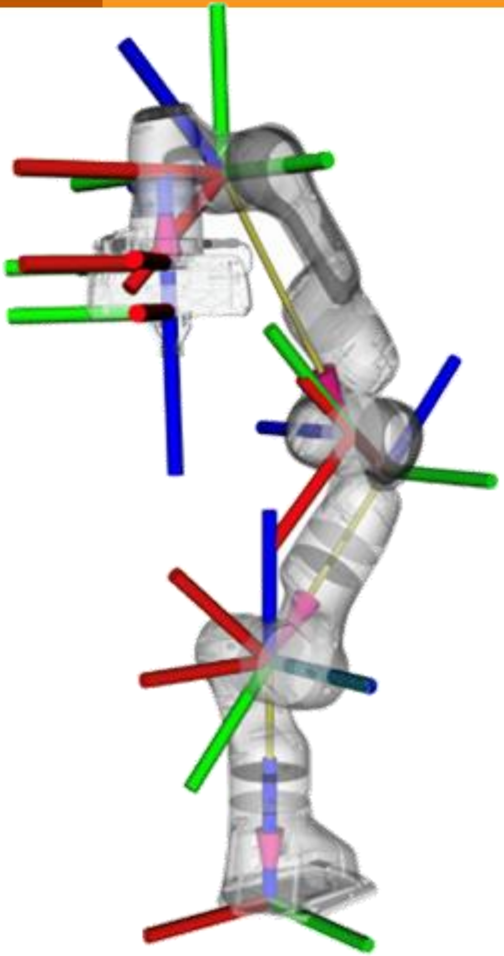
1. Displacement Vector → $\mathbb{R}^3$ Group

Representations of Rotations in 3D Space

1. Axis-Angle
2. Euler Angles
3. Rotation Matrix
4. Quaternions

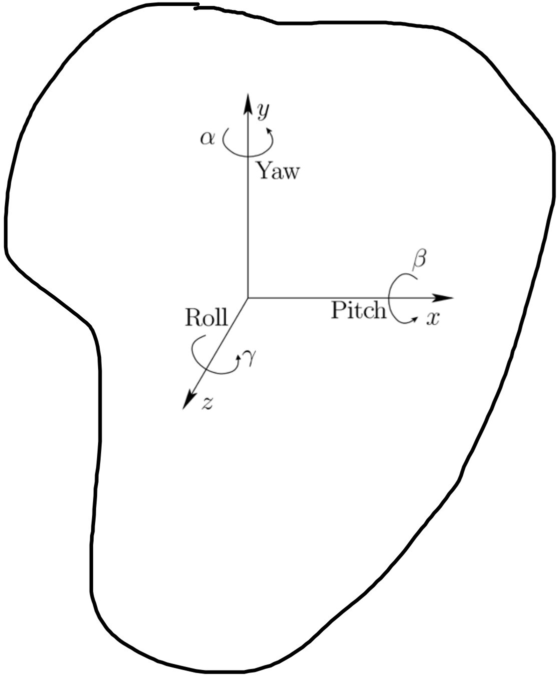
Representations of Translation+Rotation in 3D Space

1. Transformation (Homogeneous) Matrices → SE(3) Group
2. Twists → se(3) Algebra



Euler Angles

- We “attach” a frame to the rigid body
- The rotated pose of the rigid body is the result of rotating only around one axis at a time
- We can reach any orientation with sequences of rotations around three axes
- The amount of rotation and the axis we rotate around define the Euler angles



Euler Angles

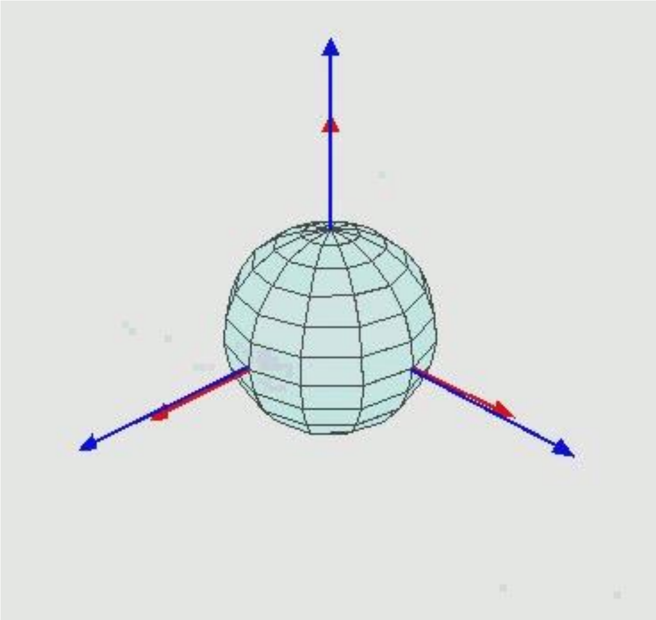
We can represent an orientation with 3 numbers

- A sequence of rotations around principal axes is called an Euler Angle Sequence

Assume we limit ourselves to 3 rotations

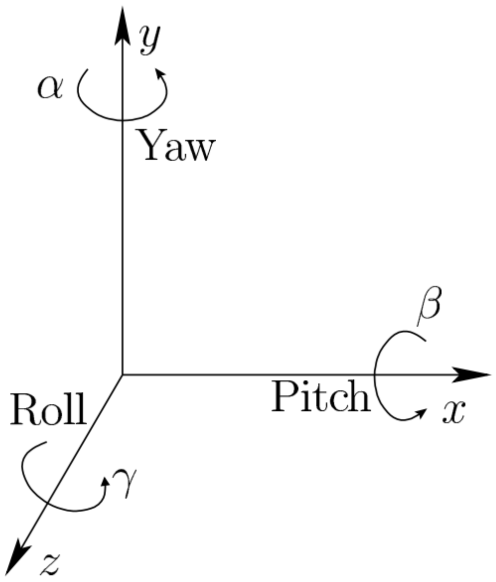
- **no successive rotations about the same axis**
- we could use any of the following 12 sequences to specify an orientation

XYZ	XZY	XYX	XZX
YXZ	YZX	YXY	YZY
ZXY	ZYX	ZXZ	ZYZ



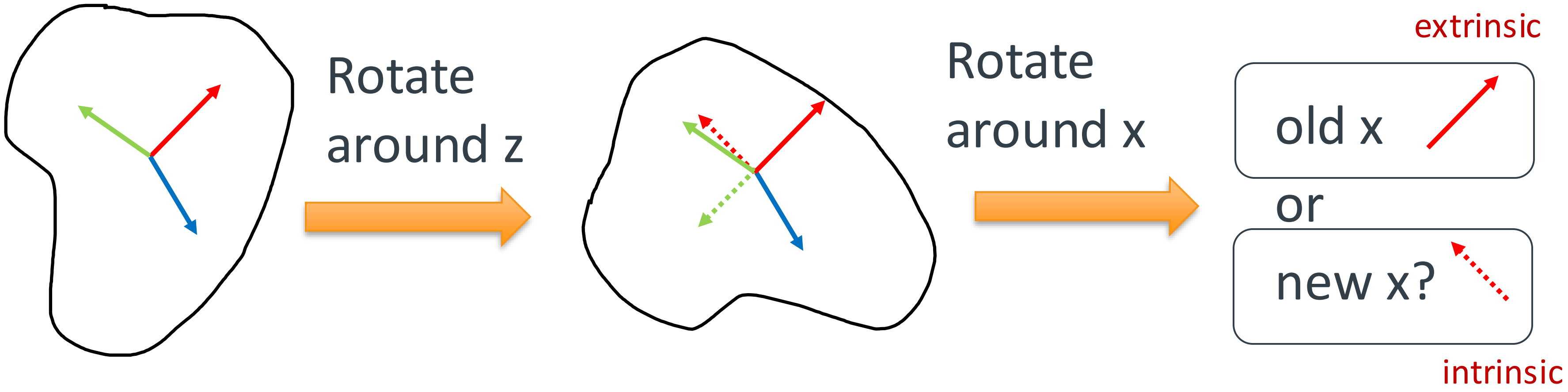
Euler Angles

- The most used Euler Angle convention
  - Roll
    - rotation about z
  - Pitch
    - rotation about x
  - Yaw
    - rotation about y
- Orientation
  - Rz(roll) Rx(pitch) Ry(yaw)

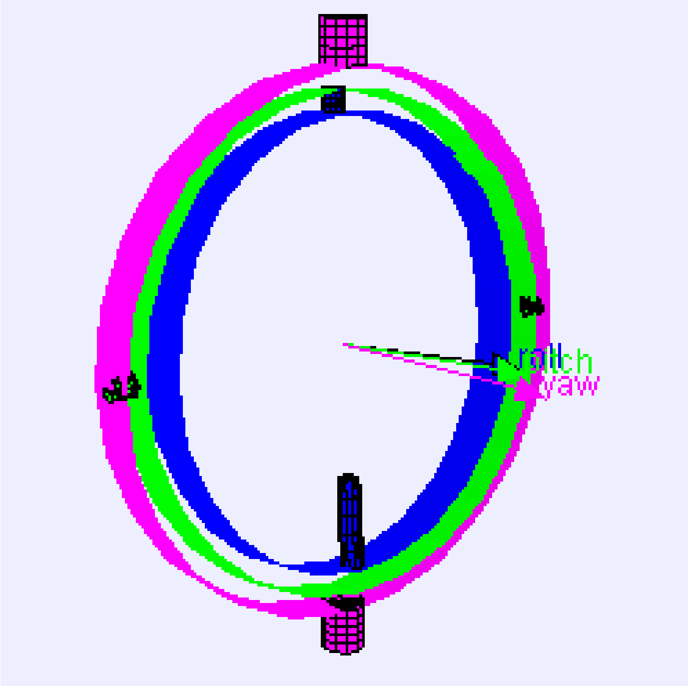


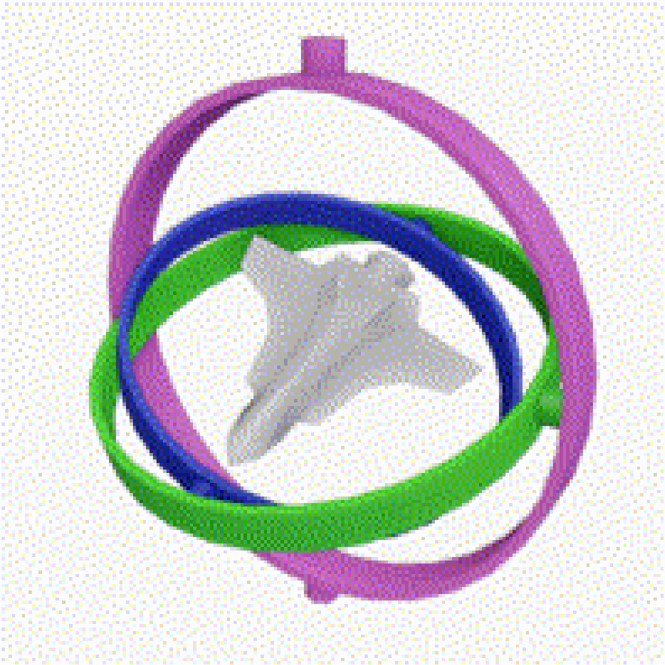
Intrinsic vs Extrinsic (Euler vs Fixed) Rotations

After the first rotation, is the next rotation wrt. the moved or the fixed axes?



Problem: Gimbal Lock





Properties of Euler Angles. They are…

Euler angles can generate any possible orientation in 3D. GOOD!

Euler angles are used in a lot of applications…they are intuitive. GOOD!

They are compact…requiring only 3 numbers*. GOOD!

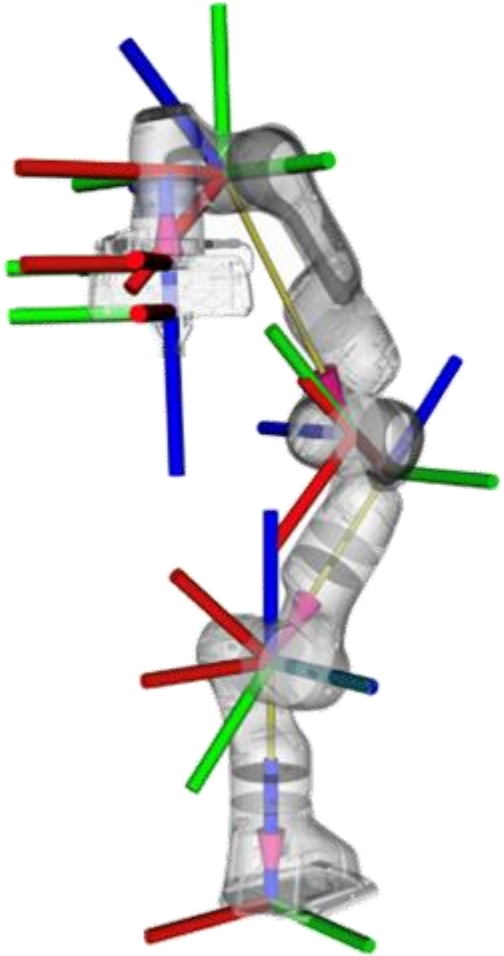
Ambiguous: different triples can be same orientation. BAD!

They do not interpolate in an obvious way (very bad in robotics). BAD!

They can suffer from Gimbal lock. BAD!

They cannot be directly applied to “move” points. BAD!

Conversion to/from a matrix requires several trig operations. BAD!



Representations of Translations in 3D Space

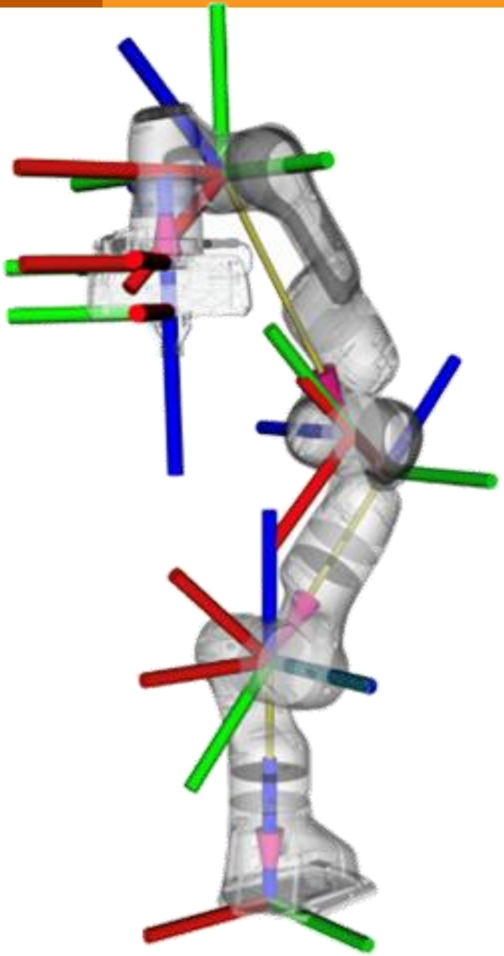
1. Displacement Vector → $\mathbb{R}^3$ Group

Representations of Rotations in 3D Space

1. Axis-Angle
2. Euler Angles
3. Rotation Matrix
4. Quaternions

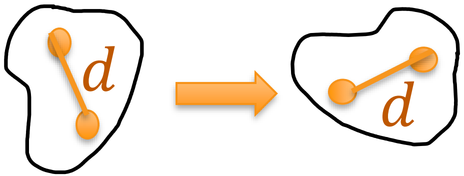
Representations of Translation+Rotation in 3D Space

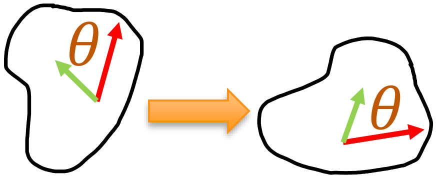
1. Transformation (Homogeneous) Matrices → SE(3) Group
2. Twists → se(3) Algebra



Rotations (and rigid body motions) as Linear Transformations

- A rotation (or any rigid body motion) is a linear transformation
- Reminder: a linear transformation is a mapping between two vector spaces that preserves the operations of vector addition and scalar multiplication
- Preserves vector addition?
  - Distances between points are preserved
- Preserves vector multiplication?
  - Angles between vectors are preserved



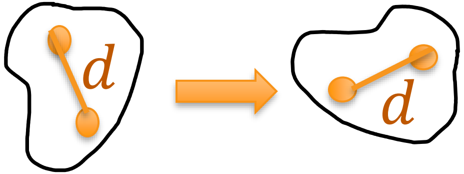


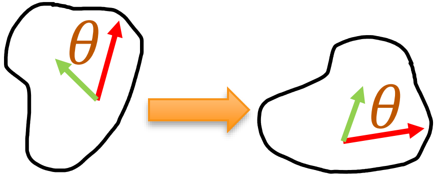
Rotations (and rigid body motions) as Linear Transformations

- If a rotation is a linear transformation there must exist a matrix such as that

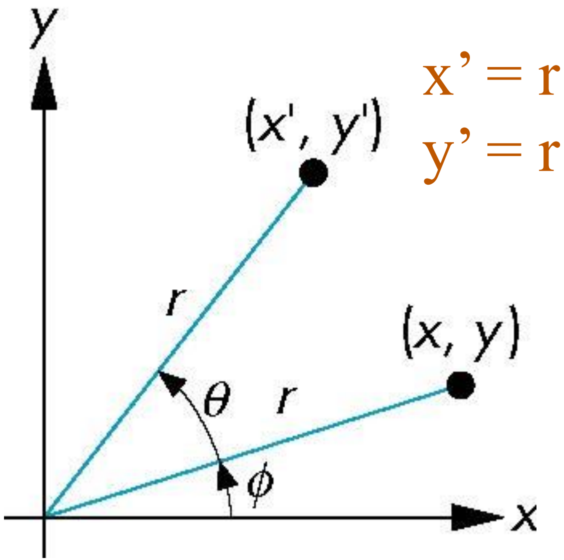
  If $p \in \mathbb{R}^3 \rightarrow p' = \mathrm{R} \cdot p$

  where R is a matrix that represents the rotation





Example: Rotation Matrix in 2D



$x' = r\cos(\phi + \theta)$
$y' = r\sin(\phi + \theta)$

$x = r\cos\phi$
$y = r\sin\phi$

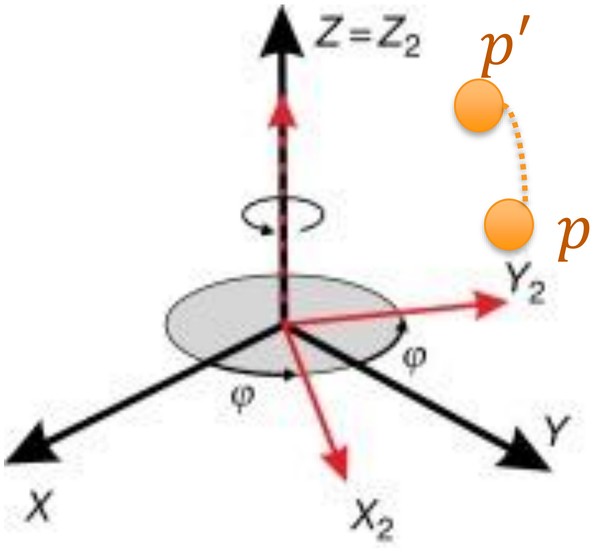
$$c_{\phi\theta} = c_\phi c_\theta - s_\phi s_\theta$$
$$s_{\phi\theta} = s_\phi c_\theta + s_\theta c_\phi$$

$$\boxed{\begin{aligned} x' &= x\cos\theta - y\sin\theta \\ y' &= x\sin\theta + y\cos\theta \end{aligned}}$$

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} c_\theta & -s_\theta \\ s_\theta & c_\theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

$$p' = \mathrm{R} \cdot p$$

Extending to Rotation Matrix in 3D for rotation around z

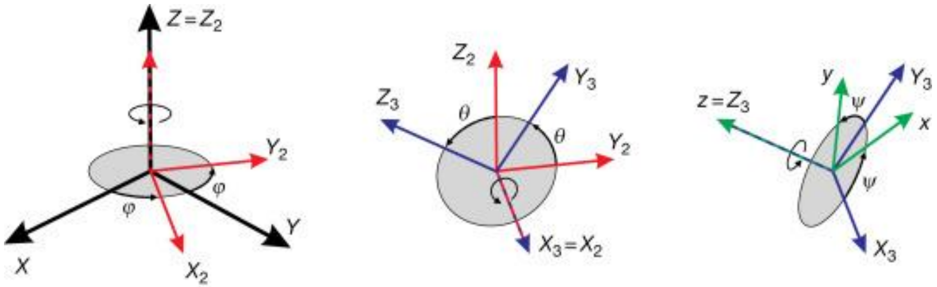


$$x` = x \cos\theta - y \sin\theta$$
$$y` = x \sin\theta + y \cos\theta$$
$$z` = z$$

$$\mathbf{p'} = \mathbf{R_Z}(\theta)\mathbf{p}$$

$$\mathbf{R} = \mathbf{R}_Z(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Rotation Matrix in 3D



$$\mathbf{R}_x(\theta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{bmatrix}$$

$$\mathbf{R}_y(\theta) = \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix}$$

$$\mathbf{R}_z(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

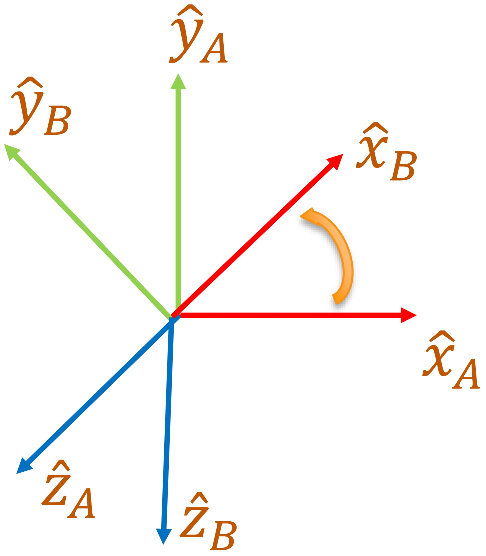
Properties of Rotation Matrices

- $R^{-1} = R^T$
- $\det(R) = 1$
- $R_1(R_2R_3) = (R_1R_2)R_3$
- ${}^A_BR = ({}^B_AR)^T$

- If a matrix R fulfills all these properties (actually, only the first 2) then, it is a rotation matrix
- We say $R \in SO(3)$, Special Orthogonal Group in 3D Space

- A Group G is a set with an operation · and with the properties:
  1. Identity: $\exists e \in G \; such \; that \; \forall a \in G, a \cdot e = a$
  2. Inverse: $a \in G \;, \exists a^{-1} such \; that \; a \cdot a^{-1} = e$
  3. Associative: $a, b, c \in G, (ab)c = a(bc)$
  4. Closure: $a, b \in G, \quad c = a \cdot b \in G$

Another way to understand/create Rotation Matrices



- Frame $\{A\}$
- Frame $\{B\}$
- Express each of the rotated unitary vectors ($\hat{x}_B$, $\hat{y}_B$ and $\hat{z}_B$) in the original coordinate frame $\{A\}$
- Use them as columns for the rotation matrix from $\{A\}$ to $\{B\}$
- $${}^B_A R = \begin{pmatrix} (\hat{x}_B)_A & (\hat{y}_B)_A & (\hat{z}_B)_A \\ \downarrow & \downarrow & \downarrow \end{pmatrix}$$

Properties of Rotation Matrices. They are…

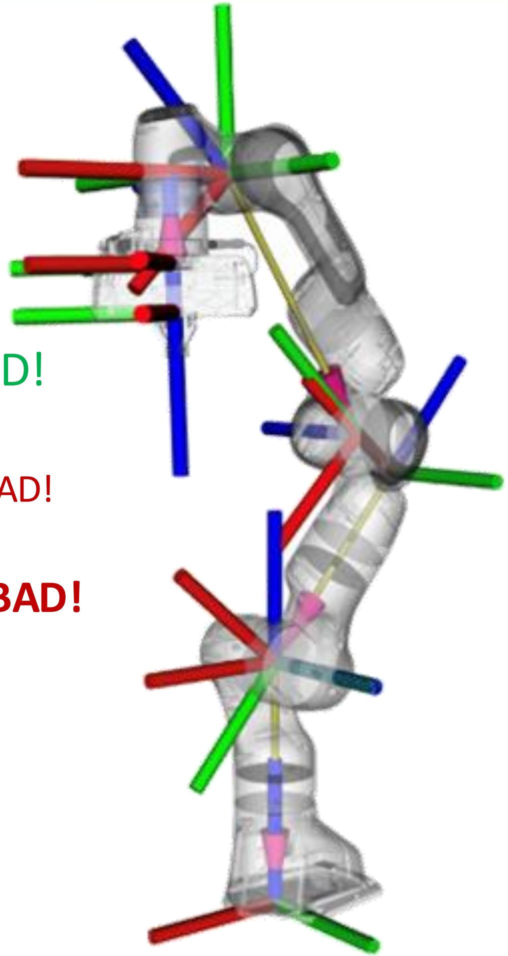
Easy to use to rotate points or frames. GOOD!

Easy to compose with other rotations (matrix multiplication). GOOD!

Not a minimal representation: 9 values for 3 degrees of freedom. BAD!

They do not interpolate in an obvious way (very bad in robotics). BAD!

Not very intuitive. BAD!



Representations of Translations in 3D Space

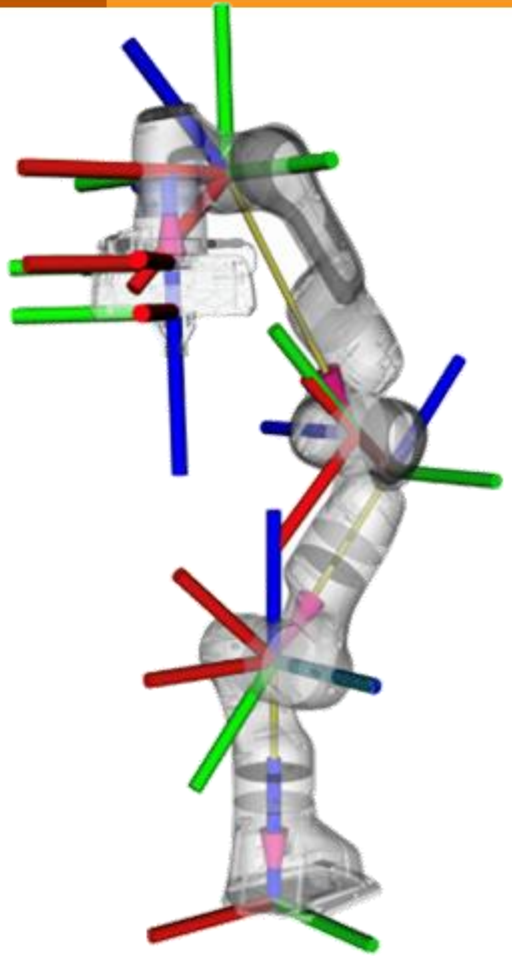
1. Displacement Vector → $\mathbb{R}^3$ Group

Representations of Rotations in 3D Space

1. Axis-Angle
2. Euler Angles
3. Rotation Matrix
4. Quaternions

Representations of Translation+Rotation in 3D Space

1. Transformation (Homogeneous) Matrices → SE(3) Group
2. Twists → se(3) Algebra



Quaternions

- Alternative to Euler Angles
- Developed by Sir William Rowan Hamilton [1843]
- Quaternions are 4-D complex numbers
  - With one real axis
  - And three imaginary axes: **i,j,k**

$$\boldsymbol{q} = q_0 + q_1\boldsymbol{i} + q_2\boldsymbol{j} + q_3\boldsymbol{k}$$
$$= (s_0, \boldsymbol{v}) = (w, x, y, z)$$

Hamilton Math Inst., Trinity College

Recall a Complex Numbers…

$z = a + i\,b$ where $i = \sqrt{(-1)}$

Can think of a complex number as having a real and an imaginary part or as a vector in two-dimensional space

$z = [a, b]$

$z1+z2 = (a1 + i\,b1) + (a2 + i\,b2) = (a1+a2) + i\,(b1+b2)$

$z1*z2 = (a1+i\,b1)*(a2+i\,b2) = (a1a2–b1b2) + i\,(a2b1+a1b2)$

Quaternions

- You can think of quaternions as an extension of complex numbers where there are "three different square roots of -1"

- q = w + *i* **x** + *j* **y** + *k* **z** where
  - $i = \sqrt{(-1)}$, $j = \sqrt{(-1)}$, $k = \sqrt{(-1)}$
  - $i*j = k$, $j*k = i$, $k*i = j$
  - $j*i = -k$, $k*j = -i$, $i*k = -j$

- You could also think of q as a value in four-dimensional space, q = [w, x, y, z]

- Sometimes written as q = [s, **v**] where s=w is a scalar and **v** is a vector in 3-space

Addition of Quaternions (easy)

$q_1 + q_2$

$= (\mathbf{w_1} + i\, \mathbf{x_1} + j\, \mathbf{y_1} + k\, \mathbf{z_1}) + (\mathbf{w_2} + i\, \mathbf{x_2} + j\, \mathbf{y_2} + k\, \mathbf{z_2})$

$= \mathbf{w_1} + \mathbf{w_2} + i\, (\mathbf{x_1} + \mathbf{x_2}) + j\, (\mathbf{y_1} + \mathbf{y_2}) + k\, (\mathbf{z_1} + \mathbf{z_2})$

q1 + q2

= [w1, **v1**] + [w2, **v2**]

= [w1+w2, **v1**+**v2**]

Multiplication of Quaternions (hard)

2 quaternions multiplied together result in a quaternion.

q1*q2 = (w1+ix1+jy1+kz1)*(w2+ix2+jy2+kz2)

$= \mathbf{w_1 w_2} + i\,\mathbf{w_1 x_2} + j\,\mathbf{w_1 y_2} + k\,\mathbf{w_1 z_2} +$

$-\mathbf{x_1 x_2} + i\,\mathbf{w_2 x_1} - j\,\mathbf{x_1 z_2} + k\,\mathbf{x_1 y_2} +$

$-\mathbf{y_1 y_2} + i\,\mathbf{y_1 z_2} + j\,\mathbf{w_2 y_1} - k\,\mathbf{x_2 y_1} +$

$-\mathbf{z_1 z_2} - i\,\mathbf{y_2 z_1} + j\,\mathbf{x_2 z_1} + k\,\mathbf{w_2 z_1}$

Recall,

q = w + *i* **x** + *j* **y** + *k* **z** where

- *i* = √(-1), *j* = √(-1), *k* = √(-1)
- i*j = k, j*k = i, k*i = j
- j*i = -k, k*j = -i, i*k = -j

Quaternion multiplication is <u>not commutative</u>

Or more succinctly stated,

$$\mathbf{qq}' = (q_0 + \mathbf{i}q_1 + \mathbf{j}q_2 + \mathbf{k}q_3)(q_0' + \mathbf{i}q_1' + \mathbf{j}q_2' + \mathbf{k}q_3')$$

$$= ss' - \mathbf{v}\cdot\mathbf{v}', s\mathbf{v}' + s'\mathbf{v} + \mathbf{v}\times\mathbf{v}'$$

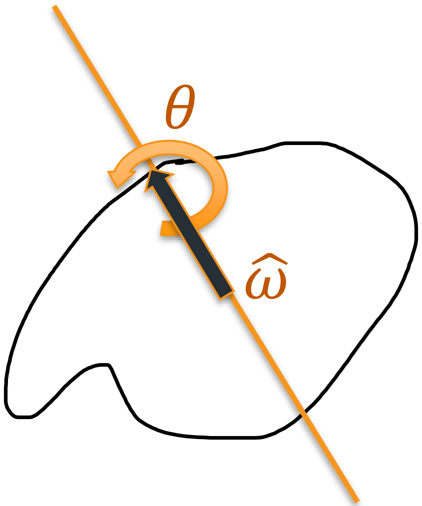
Inverse of a quaternion (easy)

$$\mathbf{q}^{-1} = \frac{(s, -\mathbf{v})}{\|\mathbf{q}\|_2^{\ 2}}$$

Ex. Find the inverse of $\mathbf{q} = 0 + 6\mathbf{i} + 8\mathbf{j} + 0\mathbf{k}$

$$\mathbf{q}^{-1} = \frac{(s, -\mathbf{v})}{\|\mathbf{q}\|_2^{\ 2}} = \frac{0 - 6\mathbf{i} - 8\mathbf{j} - 0\mathbf{k}}{\|0 + 36 + 64 + 0\|_2^{\ 2}} = 0 - .06\mathbf{i} - .08\mathbf{j} - 0\mathbf{k}$$

If the quaternion is unitary



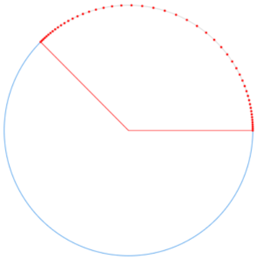
$$q = \left(w = \cos\left(\frac{\theta}{2}\right), \hat{\omega} \cdot \sin\left(\frac{\theta}{2}\right)\right)$$

Quaternions enable Spherical Interpolations (this is the big reason to use them)

- Linear Interpolation
  - between two points a and b in space

$$Lerp(t,a,b) = (1-t)a + (t)b$$

$$0 \le t \le 1$$

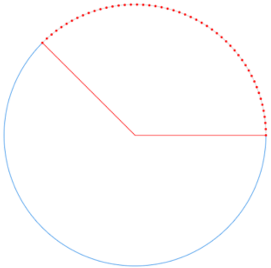


- Spherical Interpolation
  - between two quaternions a and b in space

$$Slerp(t,\mathbf{a},\mathbf{b}) = \frac{\sin((1-t)\theta)}{\sin\theta}\mathbf{a} + \frac{\sin(t\theta)}{\sin\theta}\mathbf{b}$$

$$where: \theta = \cos^{-1}(\mathbf{a}\cdot\mathbf{b})$$

Amount of rotation



Properties of Quaterions. They are…

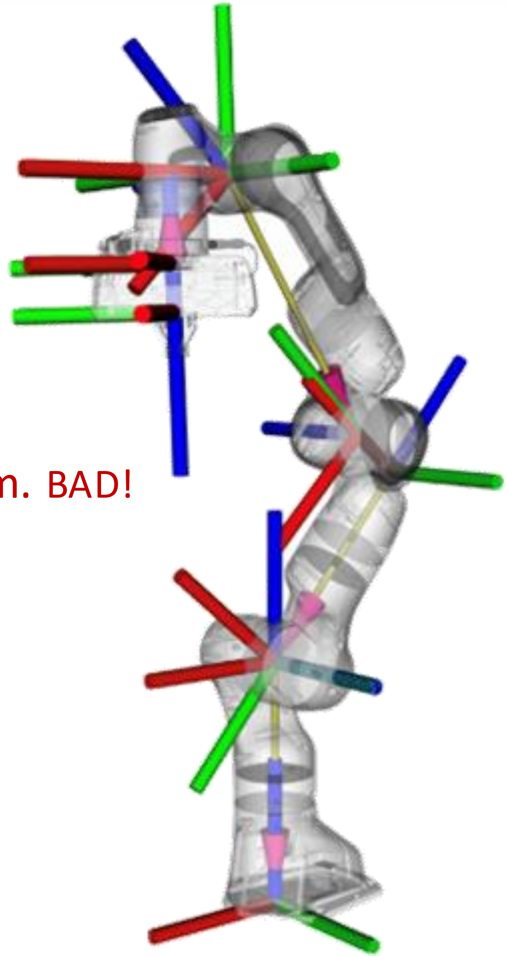
They are easy to interpolate. GOOD!

They are fine to compose. OK!

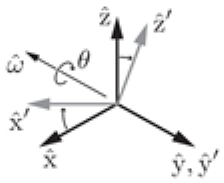
Not a minimal representation: 4 values for 3 degrees of freedom. BAD!

Hard to apply rotations to points/frames. BAD!

Not very intuitive. BAD!



Representations of Rotations



1) Rotation Matrix (direction cosine matrix)

$$R = \begin{pmatrix} \hat{x}^x_{sb} & \hat{y}^x_{sb} & \hat{z}^x_{sb} \\ \hat{x}^y_{sb} & \hat{y}^y_{sb} & \hat{z}^y_{sb} \\ \hat{x}^z_{sb} & \hat{y}^z_{sb} & \hat{z}^z_{sb} \end{pmatrix} \in SO(3)$$

2) Exponential Coordinates (Axis-angle)

$$\hat{\omega}\theta = \omega \in so(3)$$

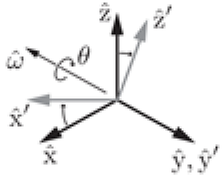
3) Euler angles

$$(\alpha, \beta, \gamma) = \text{YPR}$$

4) Quaternion

$$q = (q_w, q_x, q_y, q_z)$$

Converting between Representations (cheat sheet)



1) Rotation Matrix (direction cosine matrix)

$$R = \begin{pmatrix} \hat{x}^x_{sb} & \hat{y}^x_{sb} & \hat{z}^x_{sb} \\ \hat{x}^y_{sb} & \hat{y}^y_{sb} & \hat{z}^y_{sb} \\ \hat{x}^z_{sb} & \hat{y}^z_{sb} & \hat{z}^z_{sb} \end{pmatrix} \in SO(3)$$

$$\theta = \cos^{-1}\left(\frac{tr(R)-1}{2}\right) \qquad \begin{aligned} a_x &= (r_{32} - r_{23})/2\sin\theta \\ a_y &= (r_{13} - r_{31})/2\sin\theta \\ a_z &= (r_{21} - r_{12})/2\sin\theta. \end{aligned}$$

2) Exponential Coordinates (Axis-angle)

$$\hat{\omega}\theta = \omega \in so(3)$$

$$R = e^{[\hat{\omega}]\theta} = I + [\hat{\omega}]s_\theta + [\hat{\omega}]^2(1 - c_\theta)$$

Rodrigues formula

$$[\hat{\omega}] = \begin{bmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{bmatrix}$$

Skew Symmetric Rep.

$$R = \begin{bmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2q_1q_2 - 2q_0q_3 & 2q_1q_3 + 2q_0q_2 \\ 2q_1q_2 + 2q_0q_3 & q_0^2 - q_1^2 + q_2^2 - q_3^2 & 2q_2q_3 - 2q_0q_1 \\ 2q_1q_3 - 2q_0q_2 & 2q_2q_3 + 2q_0q_1 & q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{bmatrix}$$

$$q_0 = \cos\left(\frac{\theta}{2}\right) \quad q_1 = \hat{x}\sin\left(\frac{\theta}{2}\right) \quad q_2 = \hat{y}\sin\left(\frac{\theta}{2}\right) \quad q_3 = \hat{z}\sin\left(\frac{\theta}{2}\right)$$

$$\theta = 2\cos^{-1}(q_0)$$

$$(\hat{x}, \hat{y}, \hat{z}) = \left(\frac{q_1}{\sin(\frac{\theta}{2})}, \frac{q_2}{\sin(\frac{\theta}{2})}, \frac{q_3}{\sin(\frac{\theta}{2})}\right)$$

$$R = R(\alpha)R(\beta)R(\gamma)$$

3) Euler angles

$$(\alpha, \beta, \gamma) = \text{YPR}$$

$$q_0 = \frac{1}{2}\sqrt{tr(\mathbf{M})} \quad q_1 = \frac{m_{21} - m_{12}}{4q_0} \quad q_2 = \frac{m_{02} - m_{20}}{4q_0} \quad q_3 = \frac{m_{10} - m_{01}}{4q_0}$$

4) Quaternion

$$q = (q_w, q_x, q_y, q_z)$$

tr(M) is the trace: sum of the diagonal elements

Summary of pros and cons

1) Rotation Matrix (direction cosine matrix)

+ Operations on other geometric elements
+ Composition
+ Unique representations

- 9 elements for 3 DoF
- Interpolation

2) Exponential Coordinates (Axis-angle)

+ Minimal representation
+ Intuitive to "visualize"
+ Necessary for differential equations, integration of velocity…
- Interpolation
- Operations on other geometric elements
- Composition

3) Euler angles

+ Intuitive to "define"
+ Minimal representation
- Gimbal lock
- Composition
- Operations on other geometric elements

4) Quaternion

+ "Almost" minimal representation
+ "Almost" intuitive to "visualize"
+ Interpolation (SLERP)

- Operations on other geometric elements

What to use for what?

- Need to apply rotations to geometric elements (points, frames…)
  - Rotation matrix
- Need to concatenate/combine several rotations
  - Rotation matrix
- Need to interpolate between two orientations
  - Quaternions
- Need to define an orientation in a “manual/intuitive” way
  - Euler angles or Axis-Angle
- Need to integrate a constant angular velocity into a rotation
  - Axis-Angle

Representations of Translations in 3D Space

1. Displacement Vector → $\mathbb{R}^3$ Group

Representations of Rotations in 3D Space

1. Axis-Angle
2. Euler Angles
3. Rotation Matrix
4. Quaternions

Representations of Translation+Rotation in 3D Space

1. Transformation (Homogeneous) Matrices → SE(3) Group
2. Screw motion → se(3) Algebra