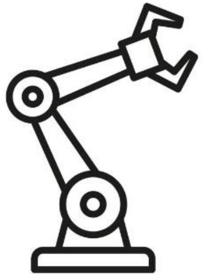
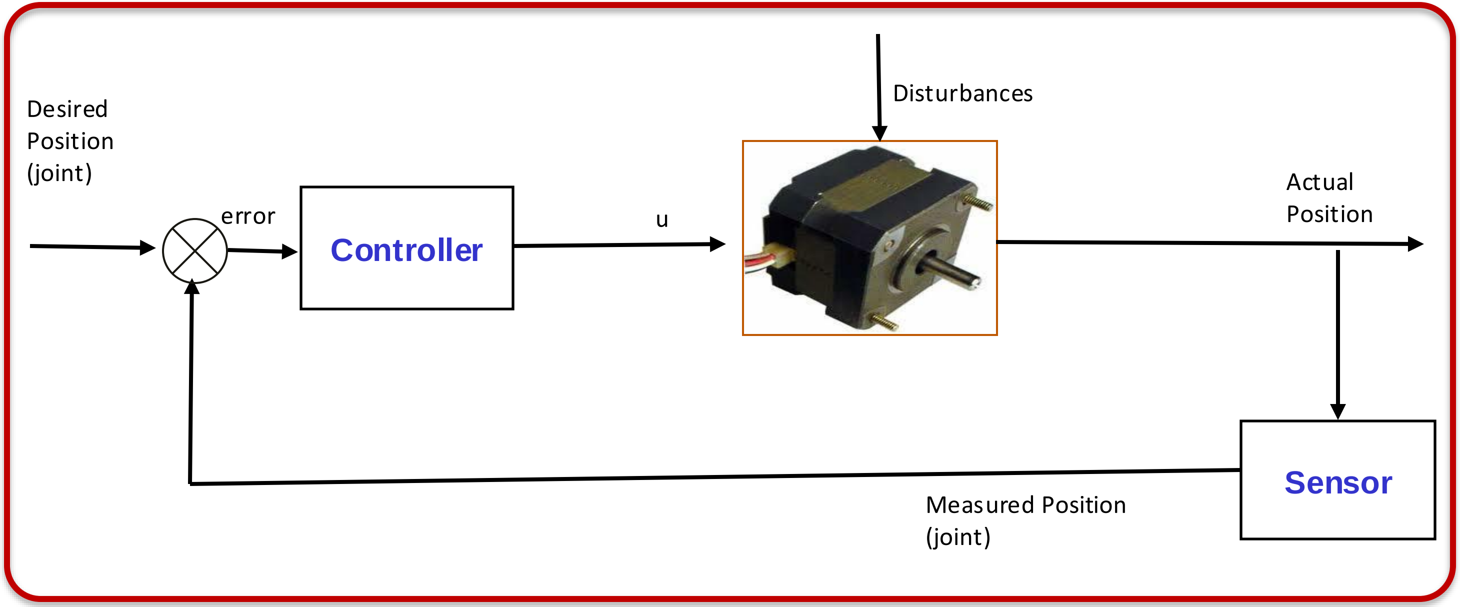
RBT350
GATEWAY TO ROBOTICS

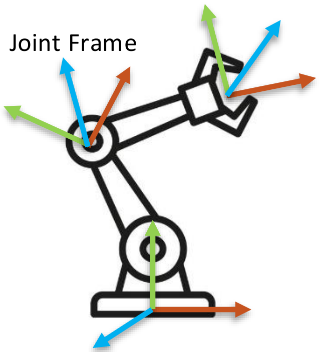
Transformations

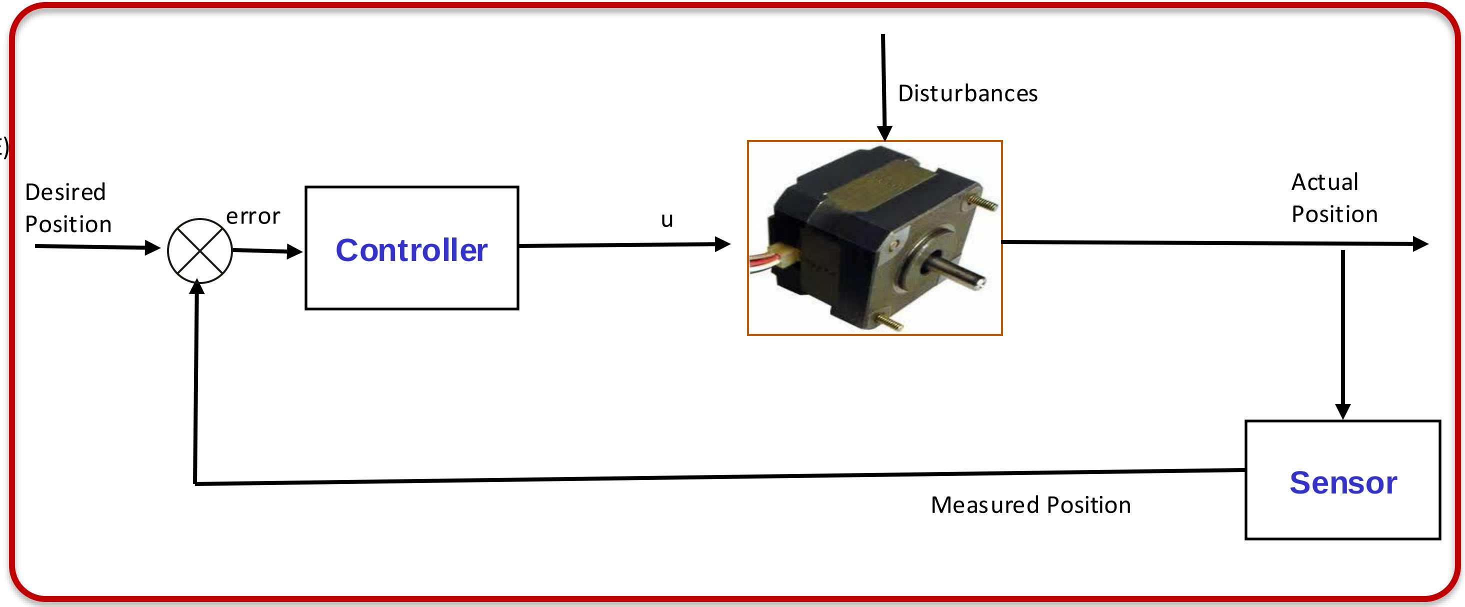
So now we can control the joints to move…





…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

Representations of Translations in 3D Space

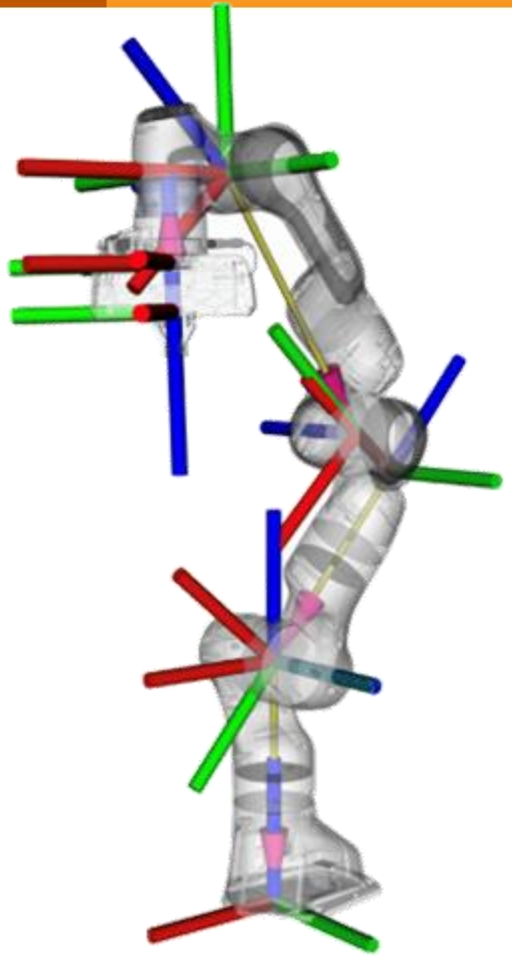
1. Displacement Vector → $\mathbb{R}^3$ Group

Representations of Rotations in 3D Space

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

Representations of Translation+Rotation in 3D Space

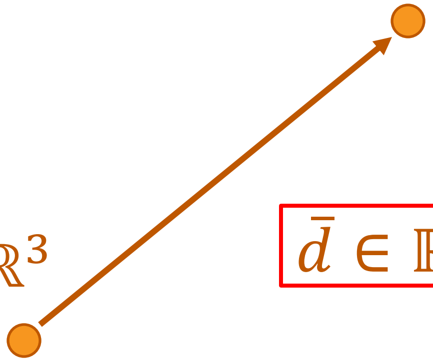
1. Transformation (Homogeneous) Matrices → SE(3) Group
2. Spatial vectors/Screw motions/Twists → se(3) Algebra



Recap: Translation

- Move a point to a new location

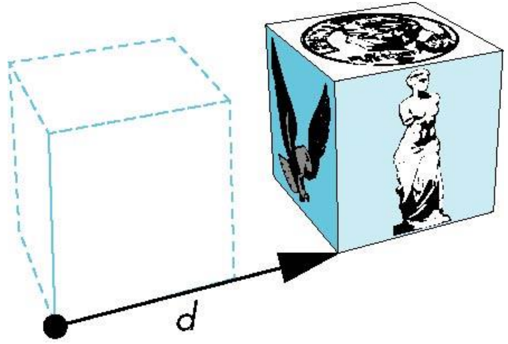
$$p' = p + \bar{d}$$



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

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

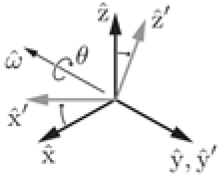
- Move an object to a new location



Every point displaced by same vector

Translation → 3 Degrees of Freedom (DoF)

Recap: 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

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

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

Transformation: Defined

- A Transformation is a rotation plus a translation.

$(R, \bar{d})$

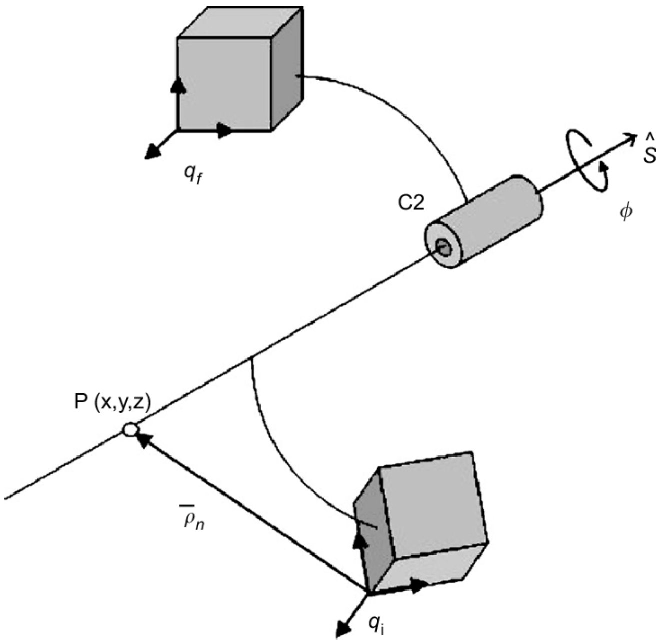
- A rotation of a point: $p' = \mathrm{R} \cdot p$
- A translation of a point: $p' = p + \bar{d}$
- A general motion of a point: $p' = \mathrm{R} \cdot p + \bar{d}$
- Several motions sequentially:

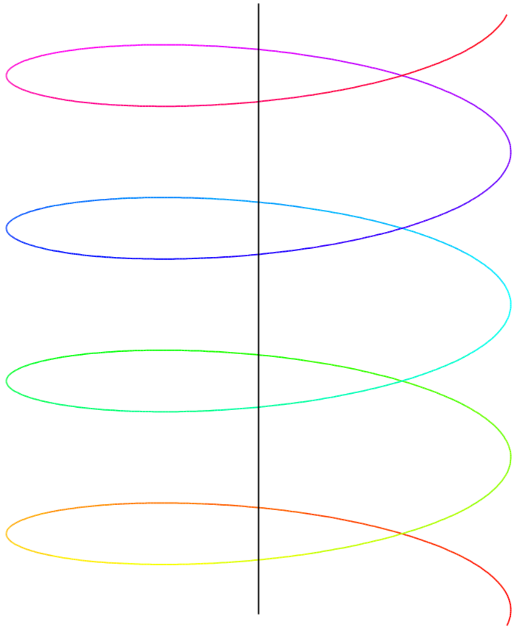
$$p' = \mathrm{R}_4(\mathrm{R}_3(\mathrm{R}_2(\mathrm{R}_1 \cdot p + \bar{d}_1) + \overline{d_2}) + \overline{d_3}) + \overline{d_4})$$

(start ”inside” work way out)

Chasles' Theorem (Screw Theory)

- Every displacement is a translation along a line and a rotation about that line.





Transformations: Representations

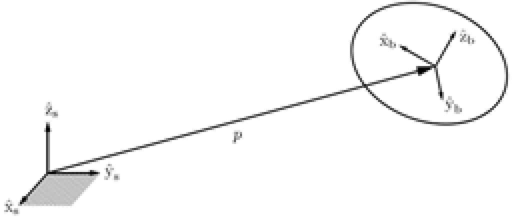
VIP

1) Homogeneous Transformation Matrix

$$T = \left(\begin{array}{c|c} R & t \\ \hline 0 & 1 \end{array} \right) \in SE(3)$$

2) Exponential Coordinates (Twist)

$$(v, \omega) \in se(3)$$



Any combination of rotation representation + translation

$$\text{e.g.} \quad (q, t) \quad \text{or} \quad (YPR, t)$$

True, but Transformations and Twists are most useful in robotics

(Homogeneous) Transformation Matrices

- If a point p is transformed with a homogenous transformation matrix, what are its new coordinates?

$$p' = T \cdot p$$

- Wait! $p \in \mathbb{R}^3$ and $T \in SE(3)$ (matrices of 4x4 with some extra properties)
  - The sizes do not match!
- We convert p into the homogeneous coordinates of p
  - Basically, we add a “1” as fourth coordinate (more complex and general, but for our purposes, that is what we will do

(Homogeneous) Transformation Matrices

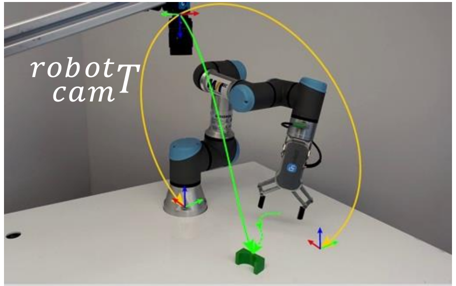
- With the homogeneous coordinates of p we get:

$$p' = T \cdot p = \begin{pmatrix} R & d \\ 000 & 1 \end{pmatrix} \begin{pmatrix} p_x \\ p_y \\ p_z \\ 1 \end{pmatrix} = \begin{pmatrix} p'_x \\ p'_y \\ p'_z \\ 1 \end{pmatrix}$$

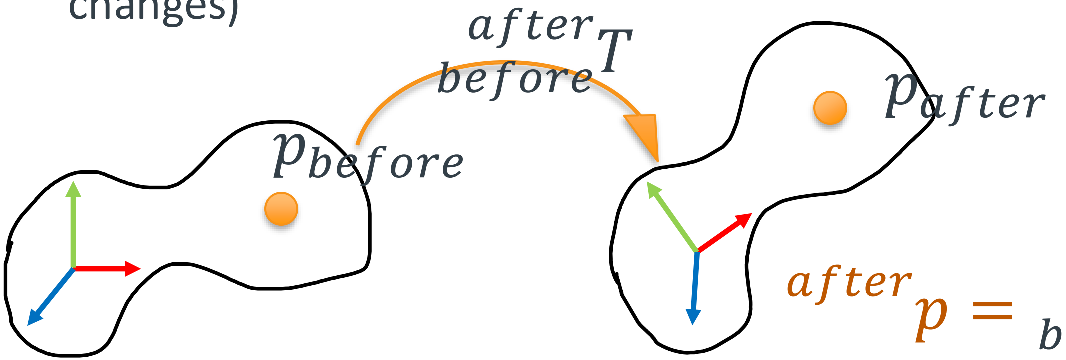
- Special cases:
  - Pure translation → $R = I$ → $p' = T \cdot p = \begin{pmatrix} I & d \\ 000 & 1 \end{pmatrix} \begin{pmatrix} p_x \\ p_y \\ p_z \\ 1 \end{pmatrix} = \begin{pmatrix} p_x + d_x \\ p_y + d_y \\ p_z + d_z \\ 1 \end{pmatrix}$
  - Pure rotation → $\mathrm{d} = 0$ → $p' = T \cdot p = \begin{pmatrix} R & 0 \\ 000 & 1 \end{pmatrix} \begin{pmatrix} p_x \\ p_y \\ p_z \\ 1 \end{pmatrix} = \begin{pmatrix} (R \cdot p)_x \\ (R \cdot p)_y \\ (R \cdot p)_z \\ 1 \end{pmatrix}$

When do you use the Transformation matrix on a point?

- Two main cases:
  - Changing the reference frame of a point (the point is the same, the coordinates change
  - Moving a point (the point changes)



$$ {}^{robot}p = {}^{robot}_{cam}T\, {}^{cam}p $$



$$ {}^{after}p = {}^{after}_{before}T\, {}^{before}p $$

(Homogeneous) Transformation Matrices

- SE(3) – The Special Euclidean Group is the set of all possible 4x4 real transformation matrices.
- Multiplying arbitrary transformations

$$\mathbf{T} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & \mathrm{d_x} \\ r_{21} & r_{22} & r_{23} & \mathrm{d_y} \\ r_{31} & r_{32} & r_{33} & \mathrm{d_z} \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \mathbf{R} & \mathbf{p} \\ 0 & 1 \end{bmatrix}$$

$$\mathbf{T} \in SE(3), \mathbf{R} \in SO(3), \mathbf{p} \in \mathbb{R}^3$$

Translation Matrix

- Express the translation using a 4x4 matrix

$$\mathbf{T} = \begin{bmatrix} 1 & 0 & 0 & \mathrm{d_x} \\ 0 & 1 & 0 & \mathrm{d_y} \\ 0 & 0 & 1 & \mathrm{d_z} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Simplifies Matrix operations

- Multiplication Preserves the Translation

$$
\begin{aligned}
{}^{2}_{0}\mathbf{T} &= {}^{1}_{0}\mathbf{T}\,{}^{2}_{1}\mathbf{T} \\
&= \begin{bmatrix} 1 & 0 & 0 & a_\mathrm{x} \\ 0 & 1 & 0 & a_\mathrm{y} \\ 0 & 0 & 1 & a_\mathrm{z} \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & b_\mathrm{x} \\ 0 & 1 & 0 & b_\mathrm{y} \\ 0 & 0 & 1 & b_\mathrm{z} \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
&= \begin{bmatrix} 1 & 0 & 0 & b_x(1) + a_x(1) \\ 0 & 1 & 0 & b_y(1) + a_y(1) \\ 0 & 0 & 1 & b_z(1) + a_z(1) \\ 0 & 0 & 0 & 1 \end{bmatrix}
\end{aligned}
$$

Regular Matrix operations preserve transformations

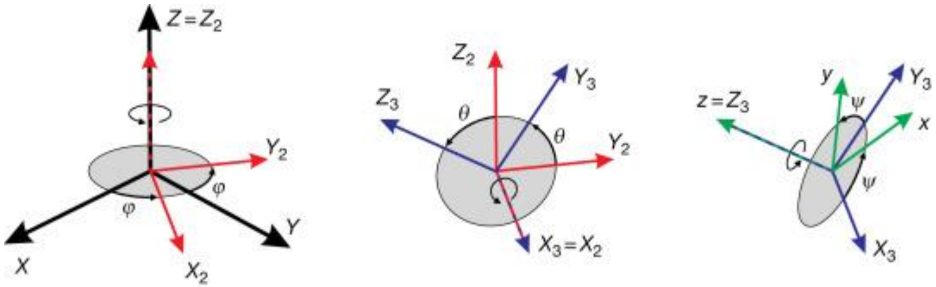
$$T = \left(\begin{array}{c|c} R & t \\ \hline 0 & 1 \end{array}\right) \in SE(3)$$

$${}^{0}_{4}T = {}^{3}_{4}T\,{}^{2}_{3}T\,{}^{1}_{2}T\,{}^{0}_{1}T$$

$$\begin{pmatrix} R_2 & d_2 \\ 000 & 1 \end{pmatrix}\begin{pmatrix} R_1 & d_1 \\ 000 & 1 \end{pmatrix} = \begin{pmatrix} R_2R1 & R_2d_1 + d_2 \\ 000 & 1 \end{pmatrix}$$

Rotation Transformations

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



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

- Multiplication Preserves the Rotation Matrix Properties

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

Properties of Transformation Matrices

- Inverse

$$\mathbf{T}^{-1} = \begin{bmatrix} \mathbf{R} & \mathbf{p} \\ 0 & 1 \end{bmatrix}^{-1}$$

$$= \begin{bmatrix} \mathbf{R}^T & -\mathbf{R}^T\mathbf{p} \\ 0 & 1 \end{bmatrix} \in SE(3)$$

- Properties

$$\mathbf{T}_1\mathbf{T}_2 \in SE(3)$$

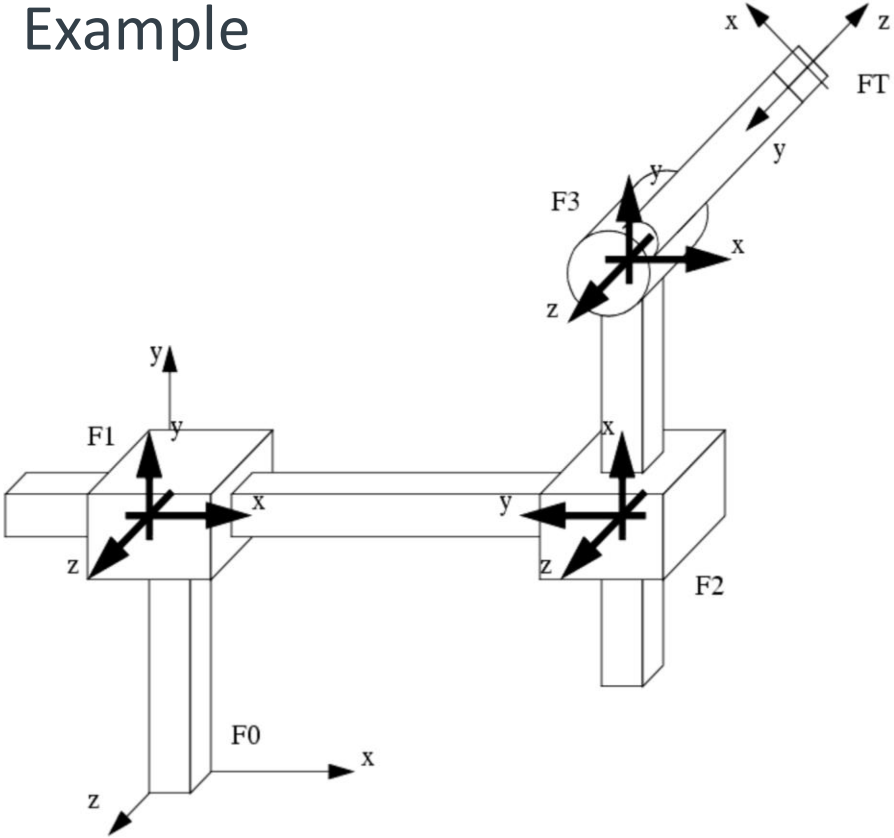
$$\left(\mathbf{T}_1\mathbf{T}_2\right)\mathbf{T}_3 = \mathbf{T}_1\left(\mathbf{T}_2\mathbf{T}_3\right)$$

$$\mathbf{T}_1\mathbf{T}_2 \neq \mathbf{T}_2\mathbf{T}_1$$

$${}^b_a\mathbf{T} = {}^a_b\mathbf{T}^{-1}$$

$${}^a_b\mathbf{T}\,{}^b_c\mathbf{T} = {}^a_c\mathbf{T}$$

Example



$$T_{0,1} = trans(0, L_1, 0)$$

$$T_{1,2} = trans(l_2, 0, 0)rot(z, 90°)$$

$$T_{2,3} = trans(l_3, 0, 0)rot(z, -90° + \theta_1)$$

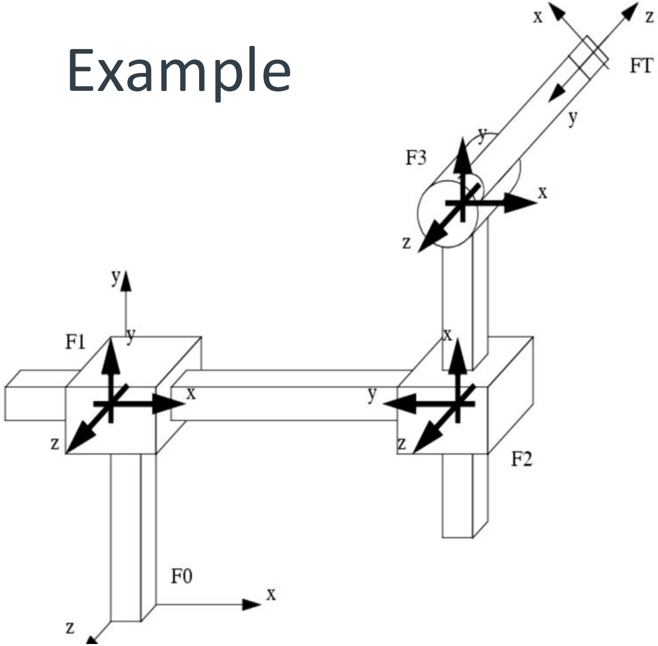
$$T_{3,T} = trans(L_4, 0, 0)rot(z, 90°)rot(x, 90°)$$

Is this a real type of robot?

https://spectrum.ieee.org/hello-robot-stretch-3

Yes.

Example



$$T_{0,1} = trans(0, L_1, 0)$$

$$T_{1,2} = trans(l_2, 0, 0)rot(z, 90°)$$

$$T_{2,3} = trans(l_3, 0, 0)rot(z, -90° + \theta_1)$$

$$T_{3,T} = trans(L_4, 0, 0)rot(z, 90°)rot(x, 90°)$$

$$T_{0,1} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & L_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_{1,2} = \begin{bmatrix} 1 & 0 & 0 & l_2 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos 90° & \sin 90° & 0 & 0 \\ -\sin 90° & \cos 90° & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_{2,3} = \begin{bmatrix} 1 & 0 & 0 & l_3 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos(-90° + \theta_1) & \sin(-90° + \theta_1) & 0 & 0 \\ -\sin(-90° + \theta_1) & \cos(-90° + \theta_1) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_{3,T} = \begin{bmatrix} 1 & 0 & 0 & L_4 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos 90° & \sin 90° & 0 & 0 \\ -\sin 90° & \cos 90° & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos 90° & \sin 90° & 0 \\ 0 & -\sin 90° & \cos 90° & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_{0,1} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & L_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_{1,2} = \begin{bmatrix} 1 & 0 & 0 & l_2 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos 90^\circ & \sin 90^\circ & 0 & 0 \\ -\sin 90^\circ & \cos 90^\circ & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_{2,3} = \begin{bmatrix} 1 & 0 & 0 & l_3 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos(-90^\circ + \theta_1) & \sin(-90^\circ + \theta_1) & 0 & 0 \\ -\sin(-90^\circ + \theta_1) & \cos(-90^\circ + \theta_1) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_{3,T} = \begin{bmatrix} 1 & 0 & 0 & L_4 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos 90^\circ & \sin 90^\circ & 0 & 0 \\ -\sin 90^\circ & \cos 90^\circ & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos 90^\circ & \sin 90^\circ & 0 \\ 0 & -\sin 90^\circ & \cos 90^\circ & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^0_2\mathbf{T} = \begin{bmatrix} 1 & 0 & 0 & l_2 \\ 0 & 1 & 0 & L_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & l_2 \\ -1 & 0 & 0 & L_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^2_3\mathbf{T} = \begin{bmatrix} 1 & 0 & 0 & l_3 \\ 0 & 1 & 0 & L_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & l_2 \\ -1 & 0 & 0 & L_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

etc.

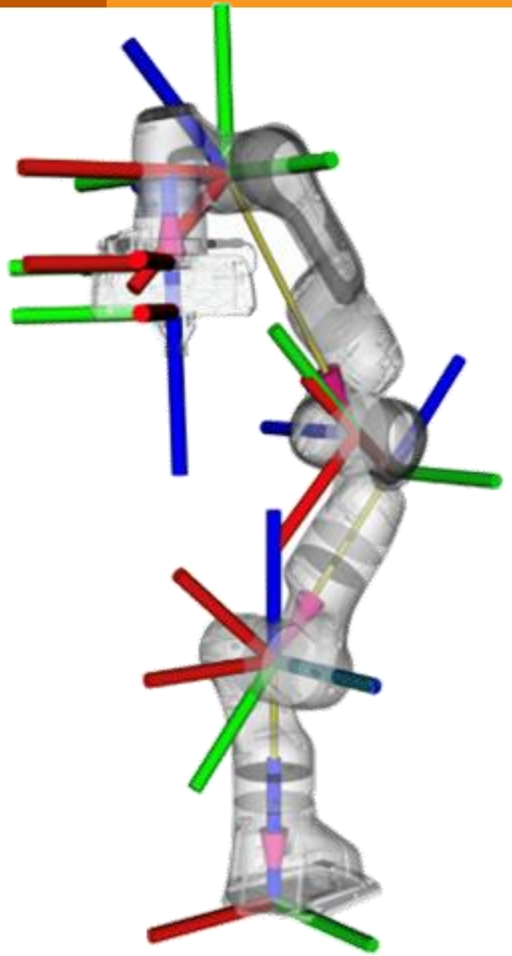
URDF – Universal Robot Description Language

```
<robot name="My 2R robot">
  <link name="base_link">
    ...
  </link>

  <link name="link_1">
    ...
  </link>

  <link name="link_2">
    ...
  </link>

  <joint name="joint_1" type="revolute">
   ...
    <parent link="base_link"/>
    <child link="link_1"/>
  </joint>

  <joint name="joint_2" type="revolute">
    ...
    <parent link="link_1"/>
    <child link="link_2"/>
  </joint>
</robot>
```

```
<link name="link_1">
  <visual>
    <origin xyz="0 0 0" rpy="0 1.5708 0" />
    <geometry>
      <cylinder length="1" radius="0.1"/>
    </geometry>
    <material name="grey">
      <color rgba="0.6 0.6 .6 1"/>
    </material>
  </visual>
  <collision>
    <origin xyz="0 0 0" rpy="0 1.5708 0" />
    <geometry>
      <cylinder length="1.0" radius="0.1"/>
    </geometry>
  </collision>
  <inertial>
    <mass value="2.0"/>
    <inertia ixx="0.2" ixy="0.0" ixz="0.0" iyy="0.6" iyz="0.0" izz="0.6"/>
  </inertial>
</link>
```

Representations of Translations in 3D Space

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



Representations of Rotations in 3D Space

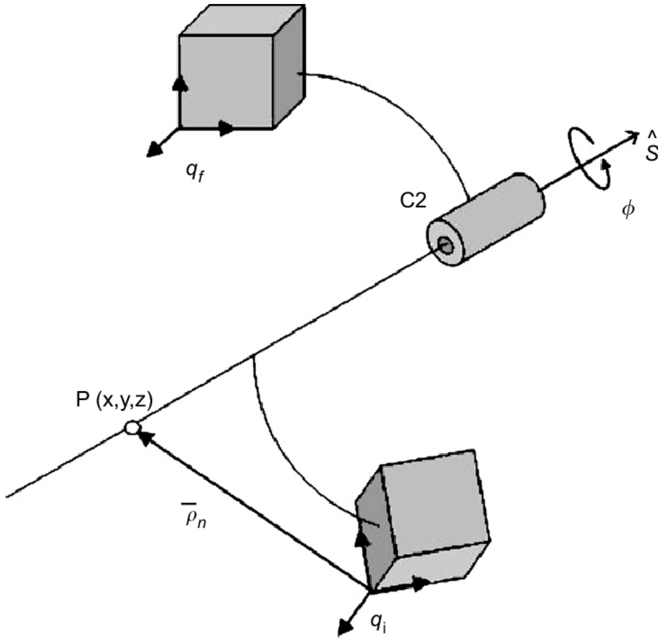
1. Axis-Angle
2. Euler Angles
3. Rotation Matrix
4. Quaternions

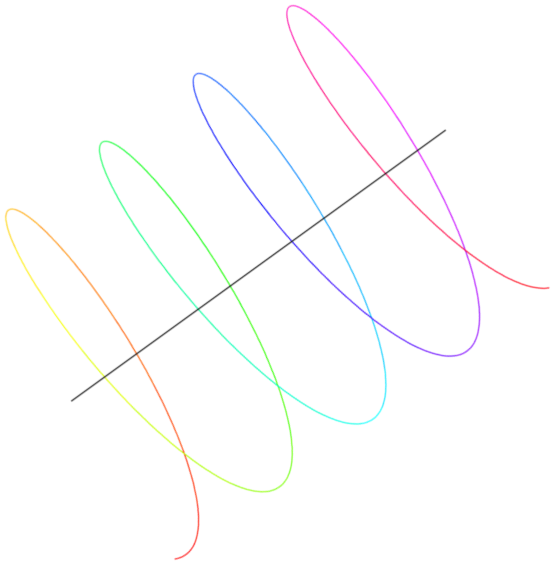
Representations of Translation+Rotation in 3D Space

1. Transformation (Homogeneous) Matrices → SE(3) Group
2. Spatial vectors/Screw motions/Twists → se(3) Algebra

Chasles' Theorem

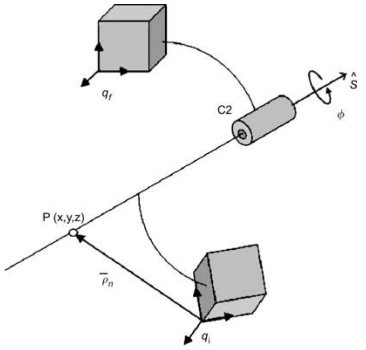
- **"Every motion is a translation along a line and a rotation about the same line"**

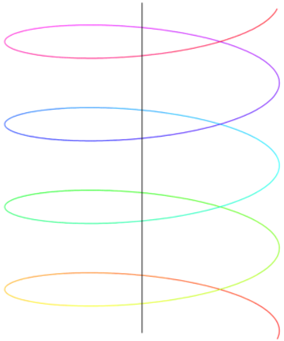


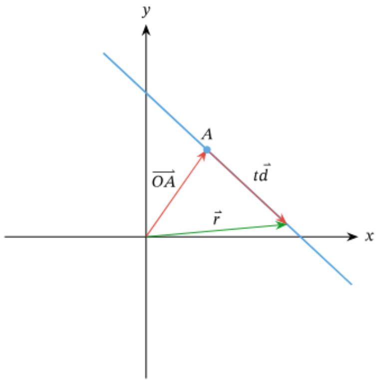


Chasles' Theorem

- **Every motion is a translation along a line and a rotation about the same line**







- To define a motion, we need to define the line and the amount of motion. Start by defining that line in standard parametric form

$$r(t) = A + t\boldsymbol{d} = \boldsymbol{p} + t\boldsymbol{q}$$

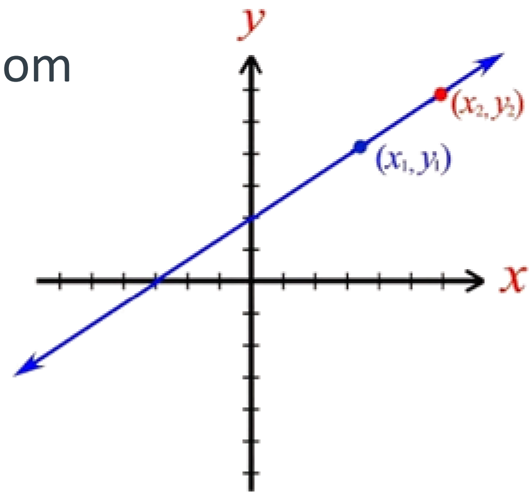
Why two equations? Same, just different symbols: Typically, Robotics texts tend to use p and q, geometers use A and d

A = $\boldsymbol{p}$ = points on the line

$\boldsymbol{d}$ = $\boldsymbol{q}$ = a direction (vector) along that line

Side note

- A given line only has 4 degrees of freedom



- One of many informal explanations:
  - A line is defined by two points (6 values), but they are constrained to that line.
  - There are many ways to define that constraint, but the simplest is to note that either point can move along the line (2 values) and still produces the same line.

But first, Plücker Coordinates

- Let’s use 6 values, but values that enable matrix operations similar to why we used transformation matrices.

Line in parametric form

$$x(t) = p + tq$$

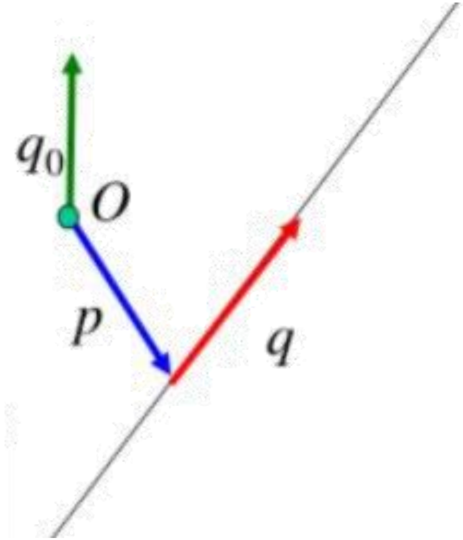
Define $q_0 \equiv p \times q$

Plucker coordinate of the line (q, q_0)

Six coordinate 4 DOFs:

$$q_0 \cdot q = 0$$

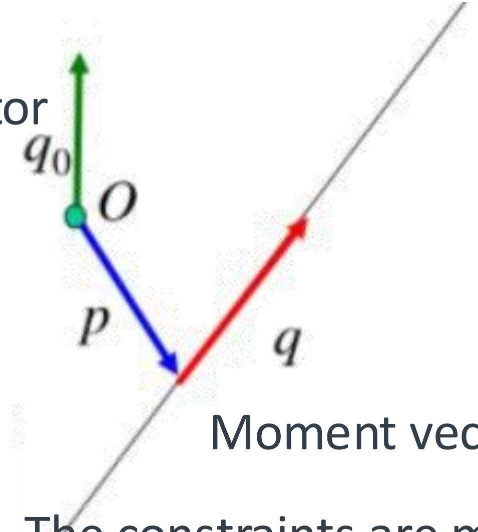
$$(q, q_0) = k(q, q_0) \text{ [scale } q \text{ by } k\text{]}$$



Plücker ($\mathbf{q}$, $\mathbf{q}_0$) Coordinates

Moment vector

- Define
  - $\mathbf{q}$ = The direction vector
    - Is parallel to the line
  - $\mathbf{q}_o$ = The moment vector
    - Is perpendicular to plane with O and the line.
- Why?
  - For two lines Plucker Coordinates are (almost) unique
  - Simplifies computation (coming soon)
  - Enables both simple algebraic and matrix algebraic solutions for
    - line-line intersections
    - line-plane intersections
    - rigid body transformations
    - Simple application of quaternion rotations



Moment vector

The constraints are mathem simple.

$$\mathbf{q}\cdot\mathbf{q}_o = 0$$

$$(\mathbf{q}, \mathbf{q}_o) = k(\mathbf{q}\cdot\mathbf{q}_o)$$

Thus, 4 degrees of freedom.

Example lines with Plücker Coordinates

- (q, q_0)
  - q is parallel to the line
  - q_0 is perpendicular to the plane with the origin of coordinates (O) and the line
  - $\frac{|q_0|}{|q|} = d$ distance to the line from O
- Special cases
  - $(q, 0)$ → Line through origin
  - $(0, q_0)$ → Line at infinity
  - $(0,0)$ → No meaning

- (q, q_0)
  - q is parallel to the line
  - q_0 is perpendicular to the plane with the origin of coordinates (O) and the line
  - $\frac{|q_0|}{|q|} = d$ distance to the line from O
- Special cases
  - $(q, 0)$ → Line through origin
  - $(0, q_0)$ → Line at infinity
  - $(0,0)$ → No meaning

Plucker Coordinate are (almost) unique

- Two lines are distinct IFF their Plucker coordinates are **linearly independent**
- Example, which two are linearly dependent (same line)?
  - q = {0, -2, -7, 7, -14, 4}
  - p = {0, 4, 14, -14, 28, -8}
  - s = {2, 1, 0, 4, 5, 6}
- Note for the correct answer, the orientation is different which is why we say (almost) unique

Exponential Coordinates of a Transformation

- We need a line (4 DoF), the amount of rotation (1 DoF) and the amount of translation (1 DoF)
- Alternatively, a line (4 DoF), the amount of ”motion” and the way rotation and translation relate (so much motion goes to rotation and so much to translation)
- Screw axis:
  - A line
  - a line (4 DoF

Screw Axis

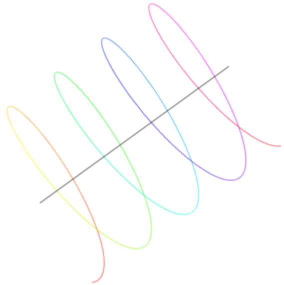
- To describe a screw axis, we need a line (4 DoF) and the way rotation and translation relate (so much translation corresponds to some rotation)
- Screw axis:
  - A line (4 DoF)
  - "Pitch" of the screw: h (1 DoF)
- Only 1 DoF "free": how much we move along the screw axis

Twists

- Pure translation: the rigid body moves with linear velocity, v
- Pure rotation: the rigid body moves with angular velocity, ω
- **"Every motion is a translation along a line and a rotation about the same line"**

- What happens if the rigid body has at the same time linear and angular velocity?
  - Twist!
  - 6D vector that express the simultaneous linear and angular velocity

- A twist is a normalized screw axis
  - The screw axis defines the direction of motion and the relationship rotation/translation
  - The twist, on top of that, tells us *how much we move per time unit*



$$\xi = (\omega, v)^T \in se(3)$$

$$(\in \mathbb{R}^6)$$

Exponential Coordinates

- Remember for rotations:
  - If a rigid body rotates with ω for a time t, it has moved $\theta = |\omega|t$ around the axis $e = \frac{\omega}{|omega|} = \hat{\omega}$ (axis-angle)
  - From axis-angle we can go into rotation matrix using the exponential transformation

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

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

Exponential Coordinates

- We can extend the same concept to full transformations (rotation and translation) and twists integrated over time
- θ amount of motion along the twist
- Exponential of a matrix is something that any math library gives us!
- Same for matrix logarithm

$$T = e^{[\hat{\xi}]\theta}$$

$$[\hat{\xi}] = \begin{bmatrix} [\widehat{\omega}] & v \\ 000 & 0 \end{bmatrix} \in \mathbb{R}^{4x4}$$

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

$$\xi = \log(T)$$

Why so many representations for pose?

1) Homogeneous Transformation Matrix

+ Operations on other geometric elements
+ Composition

- 16 elements for 6 DoF
- Interpolation

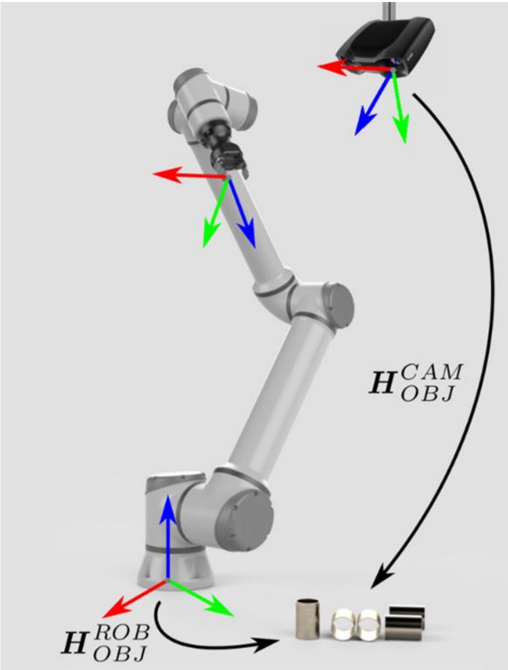
2) Exponential Coordinates (Twist)

+ Minimal representation
+ Good for optimization and iterative error minimization

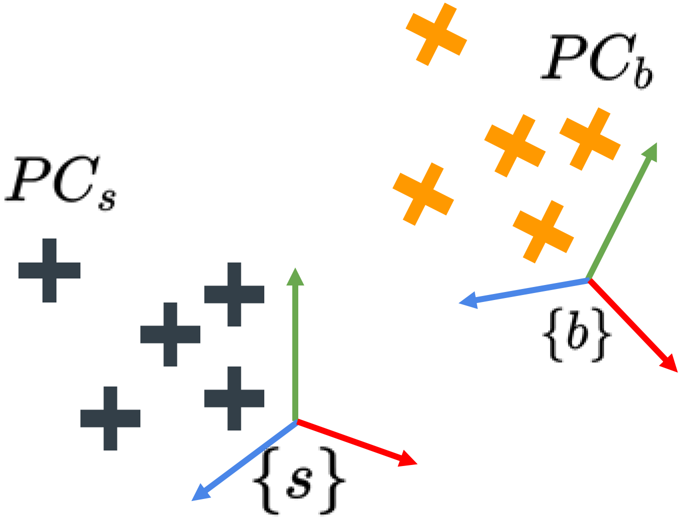
- Interpolation
- Operations on other geometric elements
- Composition

VIP: Transforming (moving, changing reference frame) of 3D points

- A very common operation in robotics
  - I know the point I want to reach in the reference frame of my camera, what is that point with respect to my robot?
  - I have a point on an object. I move the object, where is the point now?



Transforming Clouds of Points



Remember for one point:

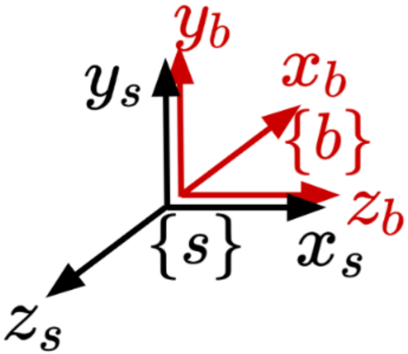
$$p' = T \cdot p = \begin{pmatrix} R & d \\ 000 & 1 \end{pmatrix} \begin{pmatrix} p_x \\ p_y \\ p_z \\ 1 \end{pmatrix} = \begin{pmatrix} p'_x \\ p'_y \\ p'_z \\ 1 \end{pmatrix}$$

1. Create a matrix with all the points in the point cloud (per column): $PC = \begin{pmatrix} {}_1p_x & \cdots & {}_Np_x \\ {}_1p_y & \cdots & {}_Np_y \\ {}_1p_z & \cdots & {}_Np_z \\ 1 & & 1 \end{pmatrix}$
2. Apply transformation on the entire point cloud:
$$PC_b = T_{sb} \cdot PC_s$$

Let's practice it!

We have a point $p = (1, 2, 3)^T$ on the surface of an object that rotates $R = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ -1 & 0 & 0 \end{pmatrix}$ What is the position of the point at the end of the rotation?

What is the rotation matrix from S to B?

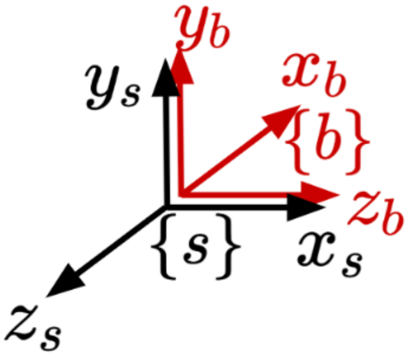


$$R_{SB} = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ -1 & 0 & 0 \end{pmatrix}$$

$$R_{SB} = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix}$$

$$R_{SB} = \begin{pmatrix} 0 & 0 & -1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix}$$

For the same rotation, what are the exponential coordinates?



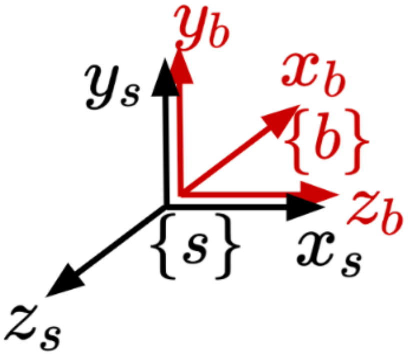
$$\omega = (0, \pi/2, 0)$$

$$\omega = (0, \sqrt{2}/2, \sqrt{2}/2) \cdot \pi/2$$

$$\omega = (0, 1, 0) \cdot \pi/2$$

$$\omega = (1, 0, 0) \cdot \pi$$

For the same rotation, what is its representation as quaternion?



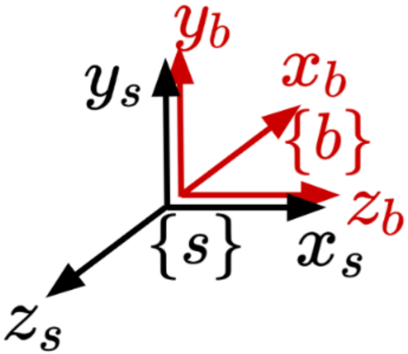
$$q = (w = 0, 1, 0, 0)$$

$$q = (w = 1, 0, 0, 0)$$

$$q = (w = \sqrt{2}/2, 0, \sqrt{2}/2, 0)$$

$$q = (w = \sqrt{2}/2, 0, 0, 1)$$

For the same rotation, what are the Euler angles (yaw, pitch, roll)?



$$\text{YPR} = (0, \pi/2, 0)$$

$$\text{YPR} = (0, \pi/2, \pi/2)$$

$$\text{YPR} = (0, \sqrt{2}/2, 0)$$

Summary

- Representative of Spatial Transformations
  - Transformation Matrices
  - Screw or Screw/Spatial Vectors, Exponential coordinates
- We can use our Super Calculator to let use define our robot in the most intuitive way and then convert internally to whatever representation we need.
- NEXT:
  - Use the information to find any part of our robot relative to any other part.
  - Find the joint angles that give us the transformation that we want!