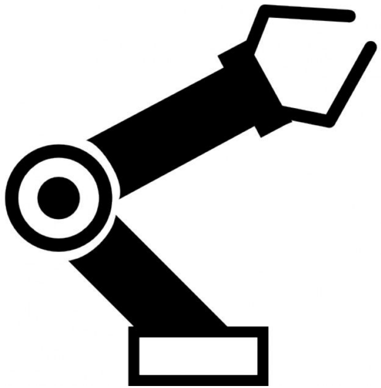


Transforms & Forward Kinematics

Linear Transformations

$$A_{3x3} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix}$$

Transformation matrix

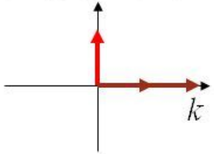
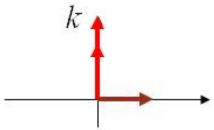
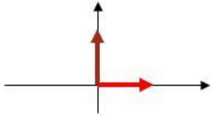
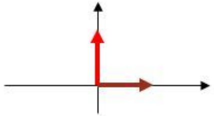
Properties:
For any 2 vectors v_1 , v_2 and constants c_1 , c_2:

1. $L(cv_1) = cL(v_1)$
2. $L(v_1) + L(v_2) = L(v_1 + v_2)$

Purpose: To transform a vector (coordinates) from one space to another

Example:

Find the standard matrix of each of the following transformations.

Horizontal Contraction & Expansion		$\begin{bmatrix} k & 0 \\ 0 & 1 \end{bmatrix}$
Vertical Contraction & Expansion		$\begin{bmatrix} 1 & 0 \\ 0 & k \end{bmatrix}$
Projection onto the *x*-axis		$\begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}$
Projection onto the *y*-axis		$\begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}$

*** Notice that the columns of the transformation matrix are the coordinate axes of the new space ***

Question:

Find the matrix that rotates a vector by 30 degrees

Combining Linear Transformations

Transformation: Scaling a vector ***v*** by a constant **k** in y-axis and then rotating by θ anticlockwise:

Transform for scaling in y by k:

$$\begin{pmatrix} 1 & 0 \\ 0 & k \end{pmatrix} \boldsymbol{v}$$

Transform for rotating by θ

$$\begin{pmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{pmatrix} \boldsymbol{v}$$

Scaling a vector *v* by a constant k in y-axis and then rotating by θ anticlockwise:

$$\begin{pmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & k \end{pmatrix} \boldsymbol{v}$$

Inversion

Let $\boldsymbol{v}$ be the original vector and $\boldsymbol{v'}$ be the vector after transforming $\boldsymbol{v}$ by a transform **A.**

$\boldsymbol{v' = \mathbf{A}\, v}$

Therefore,
$\mathbf{A}^{-1}\boldsymbol{v'} = \mathbf{A}^{-1}\mathbf{A}\boldsymbol{v} = \mathbf{v}$

Thus, $\mathbf{A}^{-1}$ allows us to get back the original vector.
Assumption: A is invertible.

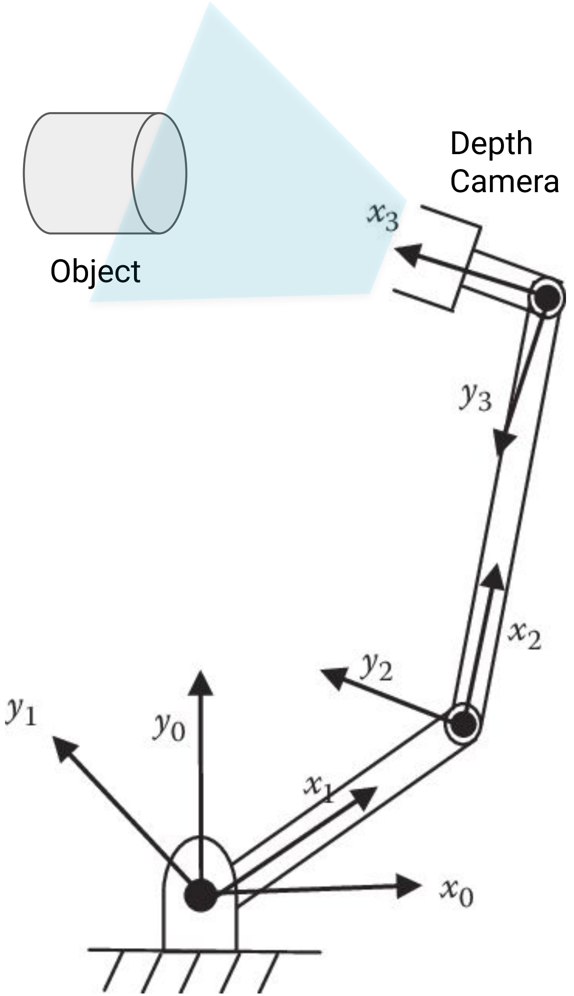
Question: What would be the inverse of a transformation that projects a vector on the y-axis?

Coordinate frames

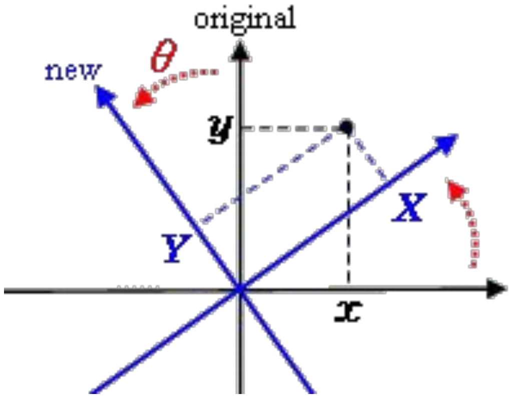
Suppose a robotic arm has a depth camera attached at the end effector. The camera detects an object at *(x,y)* with respect to itself.
What are the coordinates of the object in the global frame?

Different Coordinate Frames:

- Base frame
- Joint Frames
- End effector frame
- Global frame



Rotation of a coordinate frame



Let *(x', y')* be the coordinates of a fixed point in the new frame.

$$x' = x\cos\theta - y\sin\theta$$
$$y' = x\sin\theta + y\cos\theta$$

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

$$\boxed{R(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}}$$

2D Rotation matrix

Inverse of Rotation Matrix

Can be done using 2 methods:

1. Rotating by the same angle in the opposite direction

$$\boldsymbol{R(-\theta)} = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix}$$

2. Inverse of the original transform

$$\boldsymbol{R^{-1}(\theta)} = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix}$$

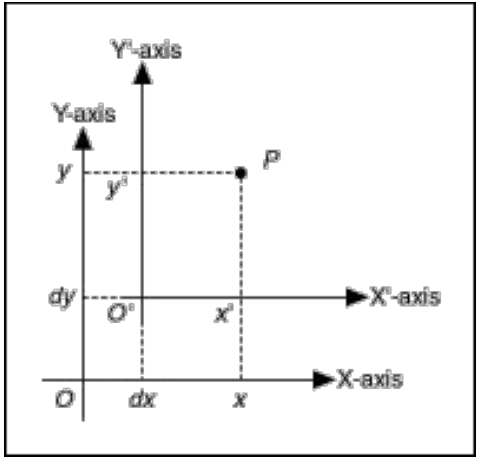
Translation of a Coordinate Frame

Translation of vector ***v*** to a new frame at *(x', y')*:

$$\mathbf{v}' = \mathbf{v} + \begin{bmatrix} x' \\ y' \end{bmatrix}$$

This is not a linear transformation → Why?

Thus, we cannot combine this with different transformations.



Requirement: A transformation matrix for translation

Solution: 'Homogeneous coordinate'

Homogeneous coordinate

A 2D vector is represented using a third row which is always 1.

i.e., *(x, y) is represented as:* $\begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$

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

Translation & Rotation with Homogeneous Coordinate

Rotation matrix in homogeneous rotation

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

Combining the translation by ***(h,k)*** and rotation by ***a,***

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos a & -\sin a & h \\ \sin a & \cos a & k \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

The Transformation Matrix

The general 2D transformation matrix can be considered to be a combination of 4 matrices

Rotation matrix	New origin
Zero vector	1

Example:

$$\begin{array}{ccc} \text{New direction of x-axis} & \text{New direction of x-axis} & \text{New origin} \end{array}$$

$$\begin{bmatrix} \cos a & -\sin a & h \\ \sin a & \cos a & k \\ 0 & 0 & 1 \end{bmatrix}$$

Transformations in 3 Dimensions

All 3 rotation matrices + the translation matrix

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

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

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

Similar to 2D, the general 3D transformation matrix can be considered to be a combination of 4 matrices

Rotation Matrix (3x3)	New origin (3x1)
Zero vector	1

Chaining Rotations

$$R = R_z(\alpha)\,R_y(\beta)\,R_x(\gamma) = \overset{\text{yaw}}{\begin{bmatrix} \cos\alpha & -\sin\alpha & 0 \\ \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix}} \overset{\text{pitch}}{\begin{bmatrix} \cos\beta & 0 & \sin\beta \\ 0 & 1 & 0 \\ -\sin\beta & 0 & \cos\beta \end{bmatrix}} \overset{\text{roll}}{\begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\gamma & -\sin\gamma \\ 0 & \sin\gamma & \cos\gamma \end{bmatrix}}$$

$$R = \begin{bmatrix} \cos\alpha\cos\beta & \cos\alpha\sin\beta\sin\gamma - \sin\alpha\cos\gamma & \cos\alpha\sin\beta\cos\gamma + \sin\alpha\sin\gamma \\ \sin\alpha\cos\beta & \sin\alpha\sin\beta\sin\gamma + \cos\alpha\cos\gamma & \sin\alpha\sin\beta\cos\gamma - \cos\alpha\sin\gamma \\ -\sin\beta & \cos\beta\sin\gamma & \cos\beta\cos\gamma \end{bmatrix}$$

Inverse of a Transform

The inverse transform, as before, is simply the inverse of the 4 x 4 transformation matrix

Question:

What would happen if the matrix is non-invertible?
Is this possible in physical transforms?

Rotation around arbitrary axes

A rotation matrix contains **9 numbers**. Are they independent?

How many numbers are needed to represent an arbitrary?

Any rotation can be represented as a rotation by angle **θ** around an arbitrary unit vector ***(l,m,n)***

Resultant rotation matrix:

$$\begin{bmatrix} ll(1-\cos\theta)+\cos\theta & ml(1-\cos\theta)-n\sin\theta & nl(1-\cos\theta)+m\sin\theta \\ lm(1-\cos\theta)+n\sin\theta & mm(1-\cos\theta)+\cos\theta & nm(1-\cos\theta)-l\sin\theta \\ ln(1-\cos\theta)-m\sin\theta & mn(1-\cos\theta)+l\sin\theta & nn(1-\cos\theta)+\cos\theta \end{bmatrix}.$$

Quaternions

Complex numbers:

$a + \boldsymbol{i}b$

where,

$i^2 = -1$

Quaternion:

$$p = p_0 + p_1\boldsymbol{i} + p_2\boldsymbol{j} + p_3\boldsymbol{k}.$$

$$\begin{aligned}
\boldsymbol{i}^2 &= \boldsymbol{j}^2 = \boldsymbol{k}^2 = \boldsymbol{ijk} = -1, \\
\boldsymbol{ij} &= \boldsymbol{k} = -\boldsymbol{ji}, \\
\boldsymbol{jk} &= \boldsymbol{i} = -\boldsymbol{kj}, \\
\boldsymbol{ki} &= \boldsymbol{j} = -\boldsymbol{ik}.
\end{aligned}$$

Representing rotations using quaternions

Let $q = q_0 + \boldsymbol{q} = q_0 + q_1\boldsymbol{i} + q_2\boldsymbol{j} + q_3\boldsymbol{k}$

Norm: qq* where,

$$q^* = q_0 - \boldsymbol{q} = q_0 - q_1\boldsymbol{i} - q_2\boldsymbol{j} - q_3\boldsymbol{k}$$

Unit Quaternion: Quaternion with norm = 1.

Any rotation can be represented as a unit quaternion

Chaining Quaternions

1. Multiplication of two unit quaternions is always a unit quaternion
2. An inverse of a unit quaternion always exists

Therefore,

Two or more **rotations** can be **combined** by simply **multiplying** their corresponding quaternions

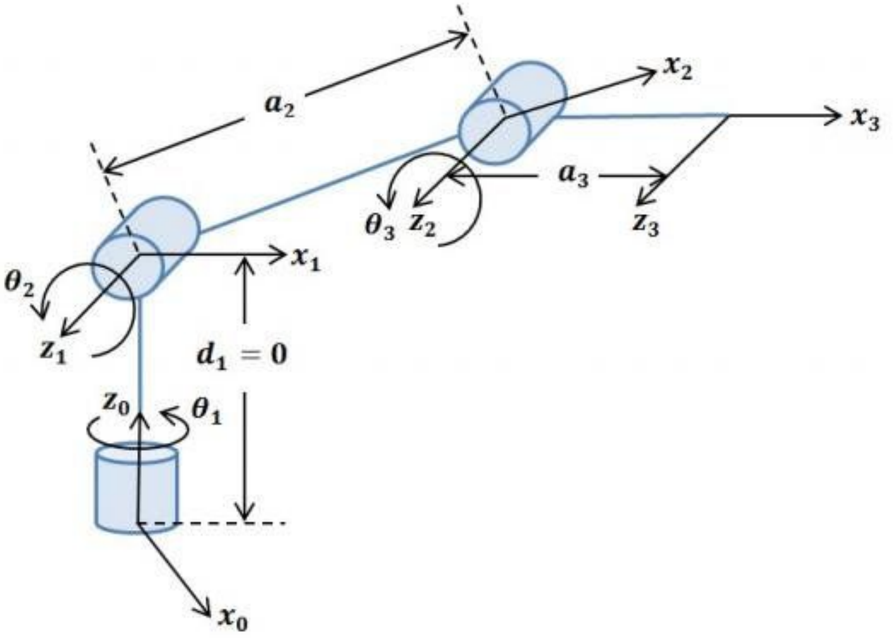
Unit Quaternion Conversions (for $q = s + \boldsymbol{i}x + \boldsymbol{j}y + \boldsymbol{k}z$)

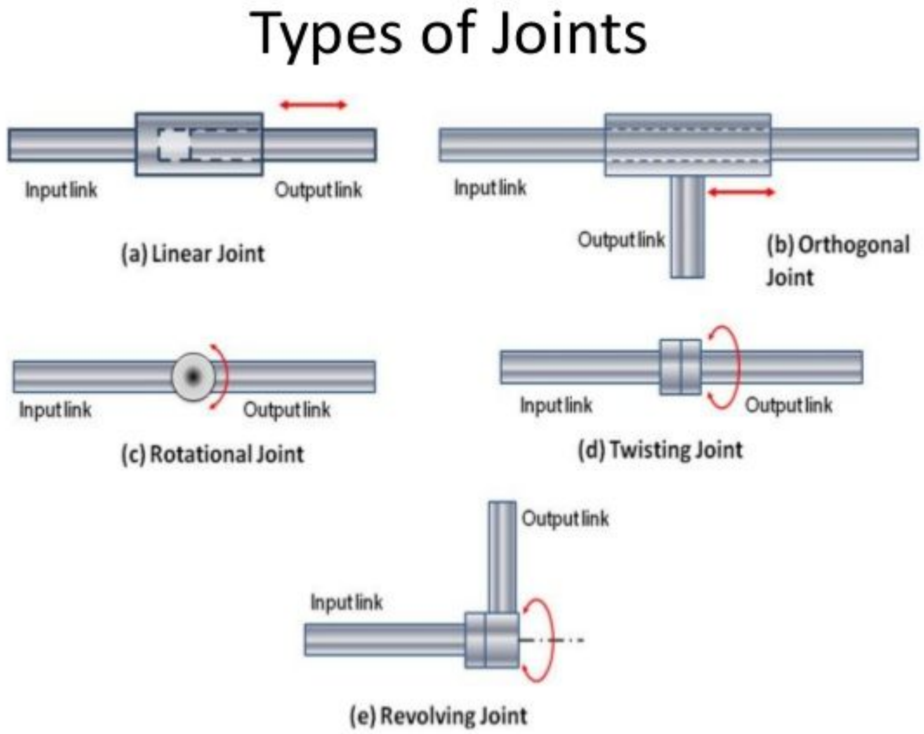
$$Rot_{[s \quad x \quad y \quad z]} = \begin{bmatrix} 1-2y^2-2z^2 & 2xy-2sz & 2xz-2sy \\ 2xy-2sz & 1-2x^2-2z^2 & 2yz-2sx \\ 2xz-2sy & 2yz-2sx & 1-2x^2-2y^2 \end{bmatrix}$$

Axis-Angle $\begin{cases} \theta = 2\cos^{-1}(s) \\ (x, y, z) = v / \|v\| \end{cases}$

Kinematic Chain (Robot Arms)

- A robot arm can be modeled as a sequence of **links** and **joints.**
- Each **joint** corresponds to a **rotation** transformation
- Each **link** corresponds to a **translation**
- As each **transformation** has a **matrix** associated with it. Hence, there exists a matrix for each pair of coordinate frames





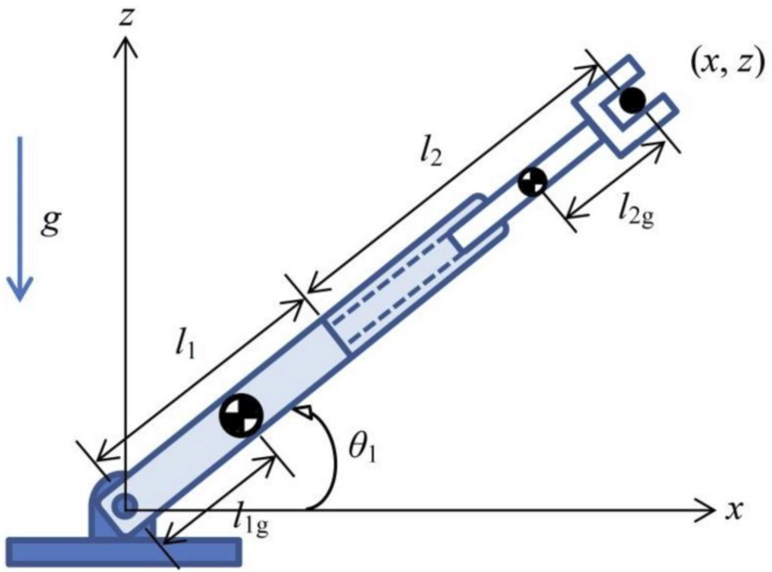
- In kinematic chains, two types of joints are used:
  - **Revolute** *(Twisting/ Revolving/ Rotational)*
  - **Prismatic** *(Linear)*
- Multi-DOF joints are modeled as a sequence of multiple joints with a zero-length link joining them

Joint State

The state of any joint can be represented using four values:

- **Revolute:**
  Axis (3 values)
  Angle (1 value)

- **Prismatic:**
  Axis (3 values)
  Displacement (1 value)



DH Parameters

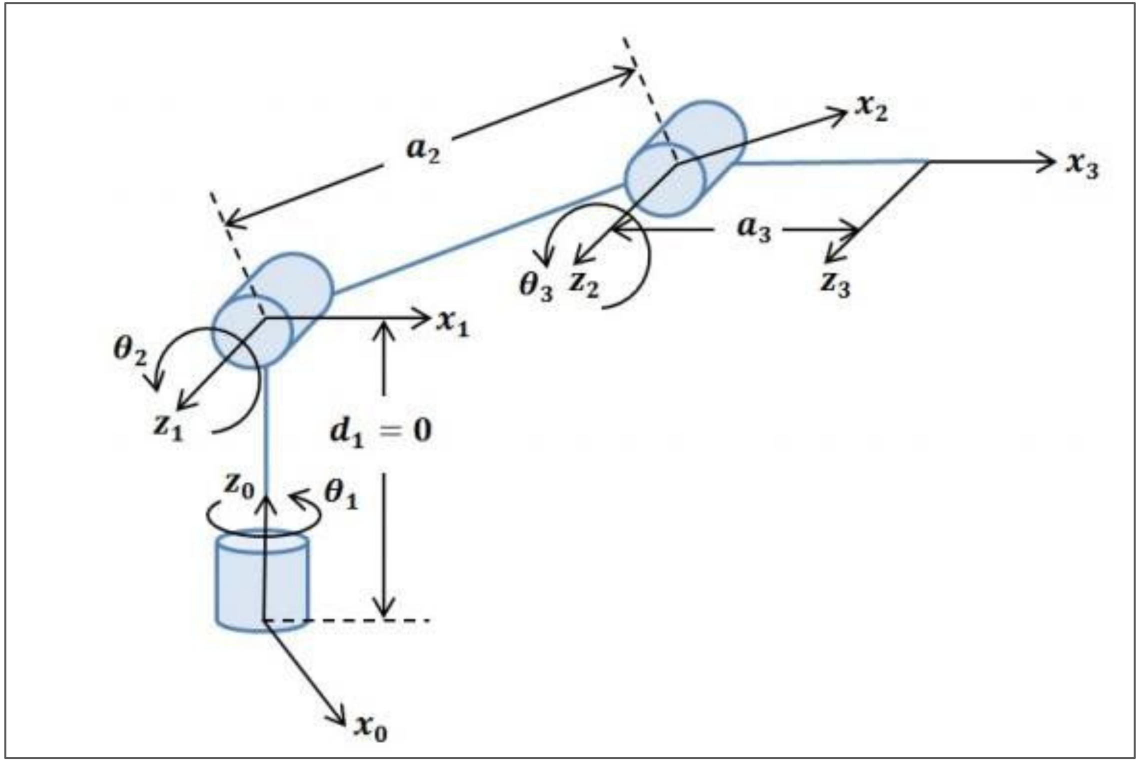
Standard form of representing the spatial relation between two joint frames in a kinematic chain, that are attached by a link:

Convention: **z** is always the axis of rotation and **x** points away from the previous joint

The relation is described using **four** parameters, as follows:

1. **d** = *offset along previous z*
2. **θ** = *angle between x-axes*
3. **r** = *length of the common normal*
4. $\boldsymbol{\alpha}$ = *angle between z-axes*

DH Parameters - Example



Forward Kinematics

Forward Kinematics: Given a kinematic chain,

1. For specific values of joints, where does the end effector go?
2. For a specific values of joints, where are all the joint frames
3. What is the transform from the base frame to all the other frames

URDF/ Xacro

Universal Robot Description Format:
XML file used by ROS to describe kinematic chains

Xacro:
Modified XML to include macros of URDFs for an easier robot description
(Can also be used to describe mobile robots)

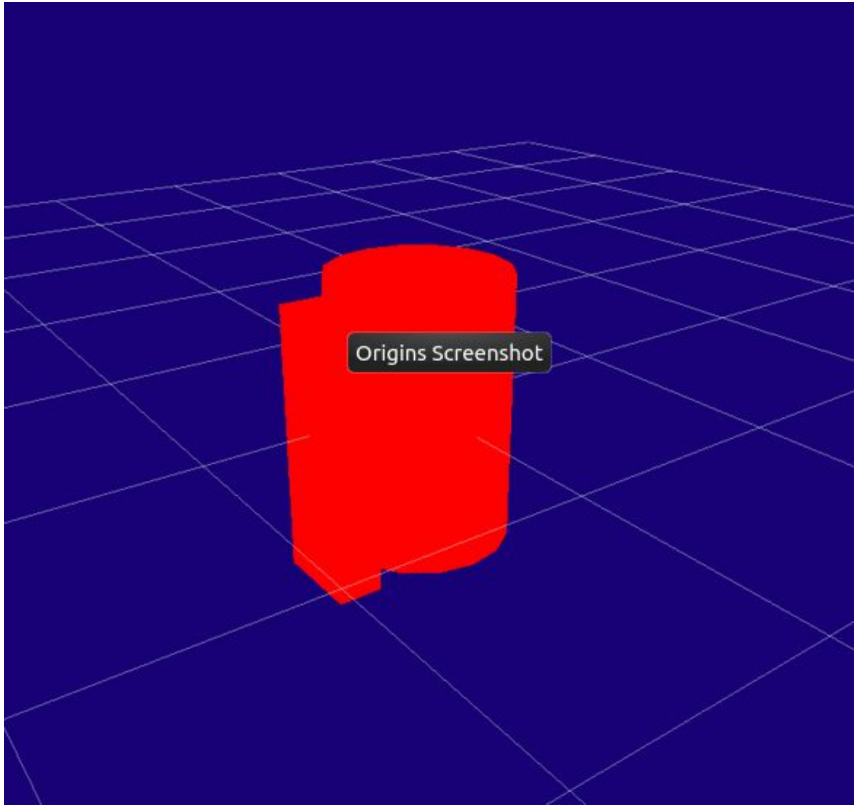
URDF example

```
<?xml version="1.0"?>
<robot name="origins">
  <link name="base_link">
    <visual>
      <geometry>
        <cylinder length="0.6" radius="0.2"/>
      </geometry>
    </visual>
  </link>

  <link name="right_leg">
    <visual>
      <geometry>
        <box size="0.6 0.1 0.2"/>
      </geometry>
      <origin rpy="0 1.57075 0" xyz="0 0 -0.3"/>
    </visual>
  </link>

  <joint name="base_to_right_leg" type="fixed">
    <parent link="base_link"/>
    <child link="right_leg"/>
    <origin xyz="0 -0.22 0.25"/>
  </joint>

</robot>
```



ROS - tf

TF (TransForms):

- tf is a ROS package that keeps a track of multiple coordinate frames over time
- It maintains the relationship between the frames in the form of a tree structure
- It allows the user to transform points, vectors, etc. at any given point of time

Resources for further learning:

ColumbiaX Robotics (EdX): Week 3

Morgan Quigley: Chapter 11 & Chapter 18