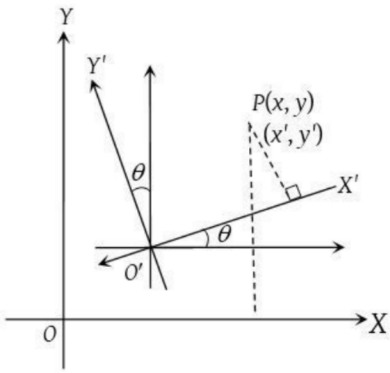
Forward Kinematics

Finding the coordinates of the end effector, given the angles of all the joints.

Graphical Transformations of Axes

- Finding transformations of one frame of reference with respect to the other.
- This concept is the basics of Forward Kinematics. Every coordinate is found with respect to one frame of reference, the Base Frame.

- https://www.aplustopper.com/transformation-axes/
- Follow this link and read about how the transformations are found
- Understand this concept thoroughly, as everything is based on it.
- This concept can be applied to 3D axes as well.



Transformations as matrices

- As you all must have studied in Math 2, you can write the transformations of axes and frames of reference in terms of matrices.

General Coordinate Transformation (rotation)

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

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

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos(x',x) & \cos(x',y) \\ \cos(y',x) & \cos(y',y) \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} \cos(x',x) & \cos(x',y) & \cos(x',z) \\ \cos(y',x) & \cos(y',y) & \cos(y',z) \\ \cos(z',x) & \cos(z',y) & \cos(z',z) \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

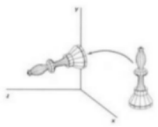
Where cos(a, b) is the cos of the angle between the 'a' and 'b' axes.

11

X-AXIS ROTATION

The equation for X-axis rotation

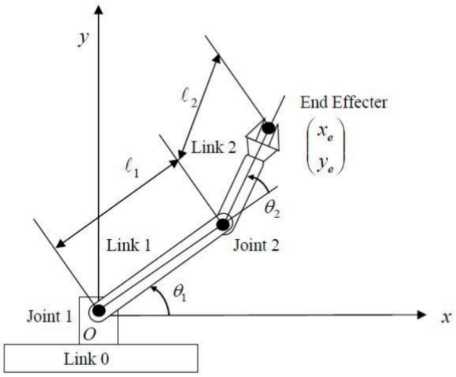
x' = x
y' = y cosθ – z sinθ
z' = y sinθ + z cosθ



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

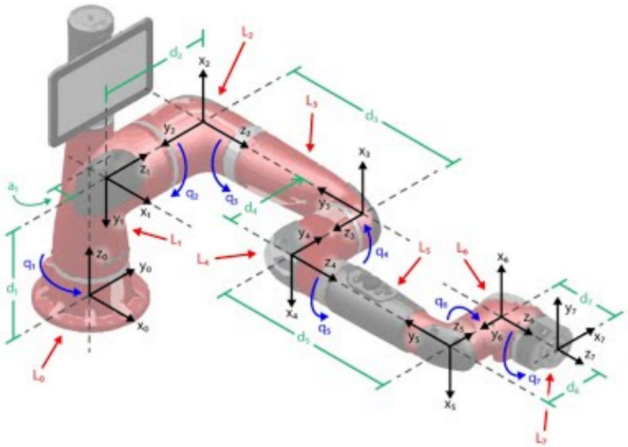
Finding Transformations of a Robotic Arm

- Now imagine this, each and every joint of a planar 2 link arm is given a frame of reference.
- The joint 1 is called the Base Joint and is given the Base Frame.
- Applying Transformations, you can easily find the coordinates of the end effector with respect to the base frame.



Finding Transformations of a Robotic Arm

- The same analogy can be applied to complex 3D arms with multiple joints and degrees of freedom.
- But finding the coordinates of the End Effector becomes way too complicated to follow this method.

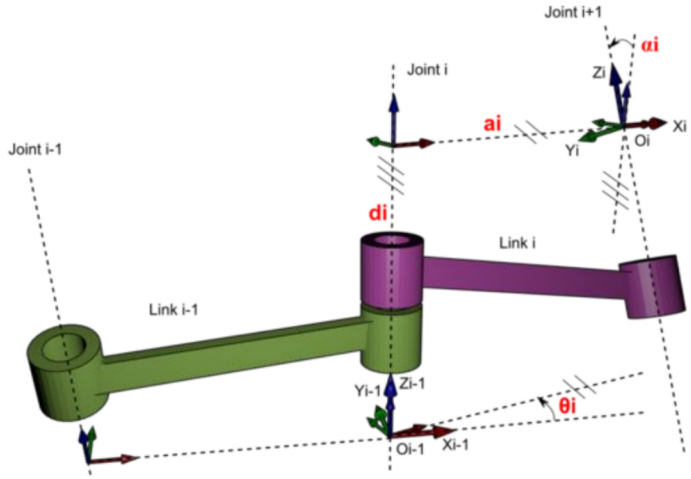


Simplifying the Task – DH Parameters

- The Denavit - Hartenberg Parameters are the 4 parameters which are associated with a particular convention for attaching the reference frames to a robotic manipulator.

- Following this system, you can calculate the forward kinematics of a robotic manipulator in much less time and much more accurately.

DH Parameters



The 4 parameters are namely-

- **d** – offset along previous z axis to the common normal.
- **θ** – angle about previous z axis, from old x to new x.
- **a** – length of the common normal.
- **α** – angle about common normal, from old z to new z.

- https://en.wikipedia.org/wiki/Denavit%E2%80%93Hartenberg_parameters

DH Matrix

- Once the parameters are determined, you can plug them in the DH matrix and calculate the transforms easily.
- If you observe closely, there are 2 parts to the matrix, the Rotation part and the Translation part – the 2 parts which determine the transformation of the reference frame.

$$
{}^{n-1}T_n = \left[\begin{array}{cccc}
\cos\theta_n & -\sin\theta_n & 0 & a_{n-1} \\
\sin\theta_n \cos\alpha_{n-1} & \cos\theta_n \cos\alpha_{n-1} & -\sin\alpha_{n-1} & -d_n \sin\alpha_{n-1} \\
\sin\theta_n \sin\alpha_{n-1} & \cos\theta_n \sin\alpha_{n-1} & \cos\alpha_{n-1} & d_n \cos\alpha_{n-1} \\
\hline
0 & 0 & 0 & 1
\end{array}\right]
$$

Using algorithms to calculate Transformations

- You can also write codes in MATLAB and python to calculate the transformation matrices, like the one showed in the picture. This one is written in MATLAB.

```
def fk_calculate(links,a,alpha,d,theta):
        F = eye(4,4)
        for i in range(0,links):
                #h = np.array([[1,2,3],[4,5,6],[7,8,9]])
                #print(theta[1])
                T= Matrix([[cos(theta[i]),-sin(theta[i])*math.cos(alpha[i]),sin(theta[i])*math.sin(alpha[i]),a[i]*cos(theta[i])],
                    [sin(theta[i]),cos(theta[i])*math.cos(alpha[i]),-cos(theta[i])*math.sin(alpha[i]),a[i]*sin(theta[i])],
                    [0,math.sin(alpha[i]),math.cos(alpha[i]),d[i]],
                    [0,0,0,1]])
                #print(F.multiply(T))
                F = F*T
                #pprint(F)
        return F[0:3,3]
```

Revision of Forward Kinematics

- Whichever method you use, you are finally calculating the transformations of the end effector with respect to the base frame.
- Using the DH parameters just provides a properly formulated method to calculate the transformations. This method has very less scope to make mistakes, as the convention to set the frames of reference is very well defined.

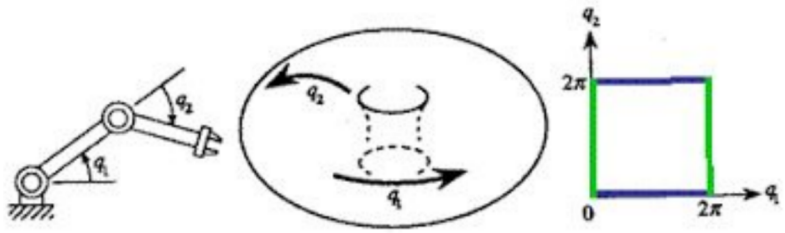
- https://blog.robotiq.com/how-to-calculate-a-robots-forward-kinematics-in-5-easy-steps

Inverse Kinematics

Finding angles subtended by all the joints, given the coordinates of the End Effector.

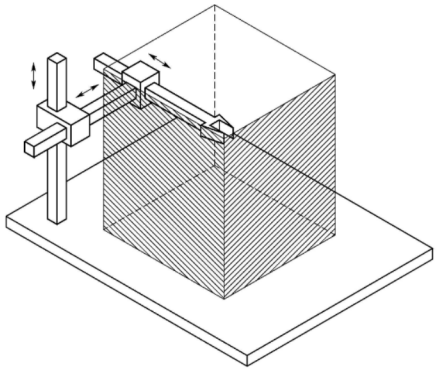
Configuration Spaces

- Configuration Space of a robotic manipulator is the plot of the *generalised coordinates*, parameters which define the configuration of a system.
- Can be defined simply as the set of the angles subtended by all the joints of the robotic manipulator.



Workspaces

- Workspace of a robotic manipulator is the set of points that can be reached by it's end effector.
- Basically, a plot of all the coordinates traced by the end effector.



Linking both the concepts

- Therefore, every point on the Configuration Space plot refers to a specific set of angles, which determine a specific point on the Workspace plot, a specific set of coordinates of the end effector.

- The configuration space and the workspace are often plot with the help of the DH parameters and the transformation matrices.

Inverse Kinematics

- This method is way more complicated than Forward Kinematics.
- Sometimes there is only 1 solution, only one point on configuration space corresponding to one point on the workspace.
- Sometimes there are multiple solutions, multiple points on the configuration space corresponding to one point on the workspace (Increases redundancy).
- And sometimes, there can be zero solutions, the given set or coordinates lies outside the workspace. Hence there will also be no such point on the configuration space.