Introduction to Robotics (Fag 3480)

Vår 2011

Robert Wood (Harward Engineeering and Applied Sciences-Basis)
Ole Jakob Elle, PhD (Modified for IFI/UIO)
Førsteamanuensis II, Institutt for Informatikk
Universitetet i Oslo
Seksjonsleder Teknologi,
Intervensjonssenteret, Oslo Universitetssykehus (Rikshospitalet)

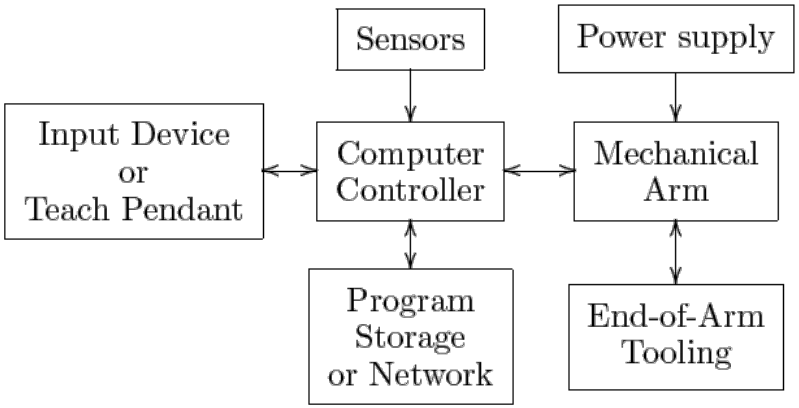
Ch. 3: Forward and Inverse Kinematics

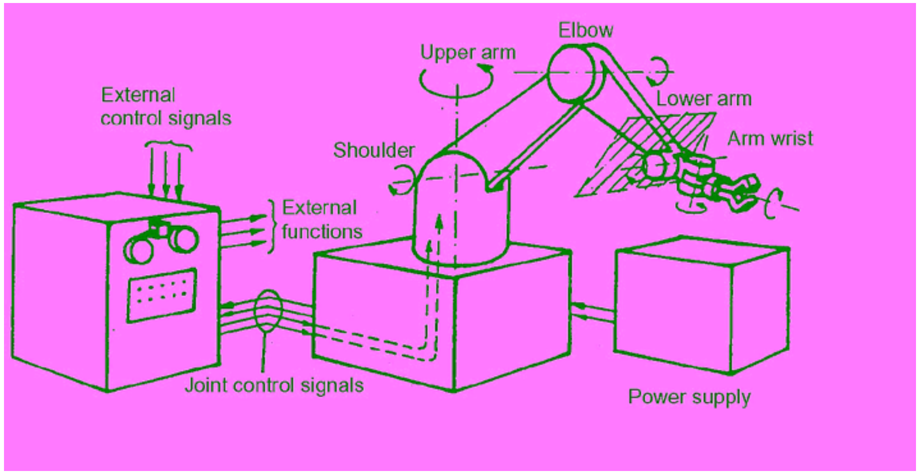
Industrial robots

High precision and repetitive tasks

Pick and place, painting, etc

Hazardous environments

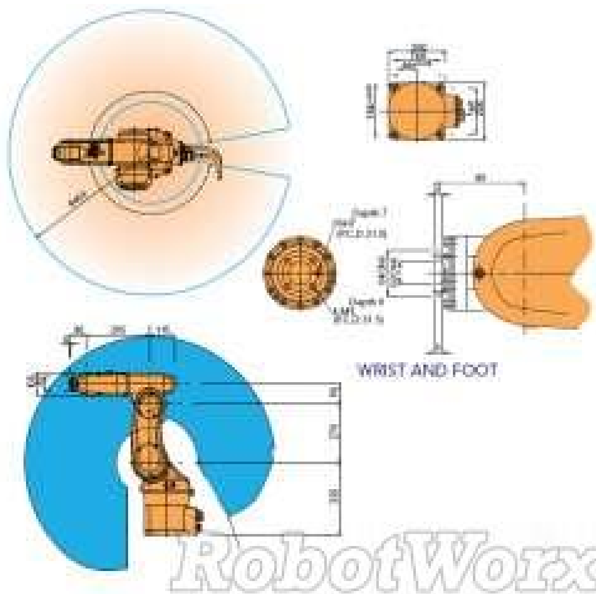




Common configurations: elbow manipulator

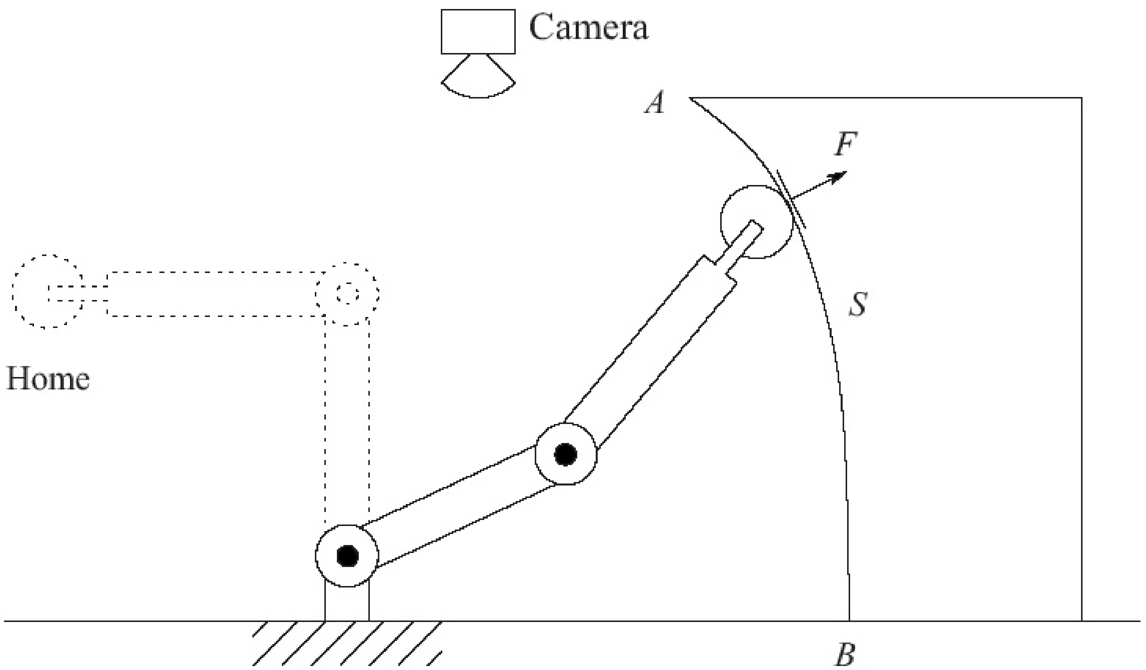
Anthropomorphic arm: ABB IRB1400 or KUKA

Very similar to the lab arm NACHI (RRR)



Simple example: control of a 2DOF planar manipulator

Move from 'home' position and follow the path AB with a constant contact force *F* all using visual feedback



Coordinate frames & forward kinematics

Three coordinate frames: ⓪ ① ②

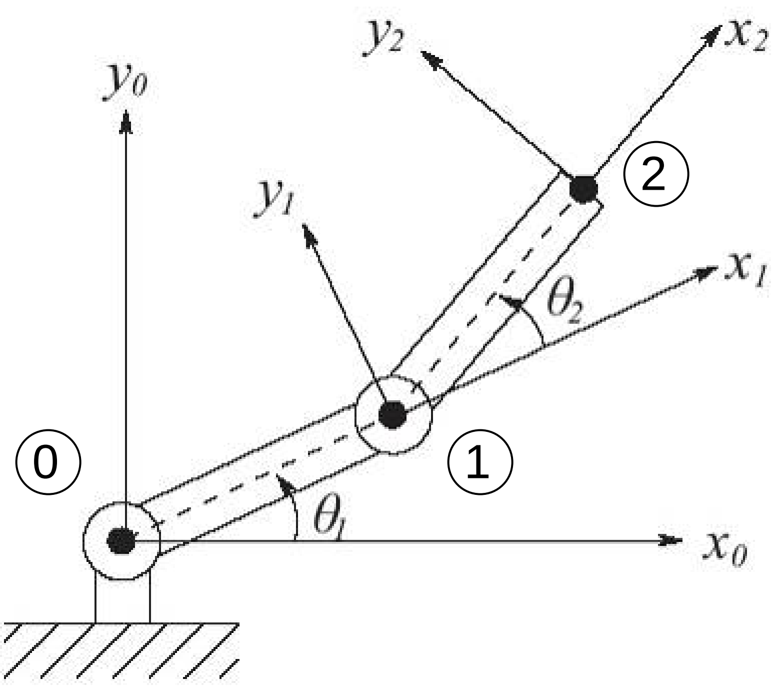
Positions:

$$\begin{bmatrix} x_1 \\ y_1 \end{bmatrix} = \begin{bmatrix} a_1\cos(\theta_1) \\ a_1\sin(\theta_1) \end{bmatrix}$$

$$\begin{bmatrix} x_2 \\ y_2 \end{bmatrix} = \begin{bmatrix} a_1\cos(\theta_1) + a_2\cos(\theta_1+\theta_2) \\ a_1\sin(\theta_1) + a_2\sin(\theta_1+\theta_2) \end{bmatrix} \equiv \begin{bmatrix} x \\ y \end{bmatrix}_t$$

$$\hat{x}_0 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \hat{y}_0 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

Orientation of the tool frame:



$$\hat{x}_2 = \begin{bmatrix} \cos(\theta_1+\theta_2) \\ \sin(\theta_1+\theta_2) \end{bmatrix}, \hat{y}_2 = \begin{bmatrix} -\sin(\theta_1+\theta_2) \\ \cos(\theta_1+\theta_2) \end{bmatrix}$$

$$R_2^0 = \begin{bmatrix} \hat{x}_2\cdot\hat{x}_0 & \hat{y}_2\cdot\hat{x}_0 \\ \hat{x}_2\cdot\hat{y}_0 & \hat{y}_2\cdot\hat{y}_0 \end{bmatrix} = \begin{bmatrix} \cos(\theta_1+\theta_2) & -\sin(\theta_1+\theta_2) \\ \sin(\theta_1+\theta_2) & \cos(\theta_1+\theta_2) \end{bmatrix}$$

Ch. 2: Rigid Body Motions and Homogeneous Transforms

Alternate approach

Rotation matrices as projections

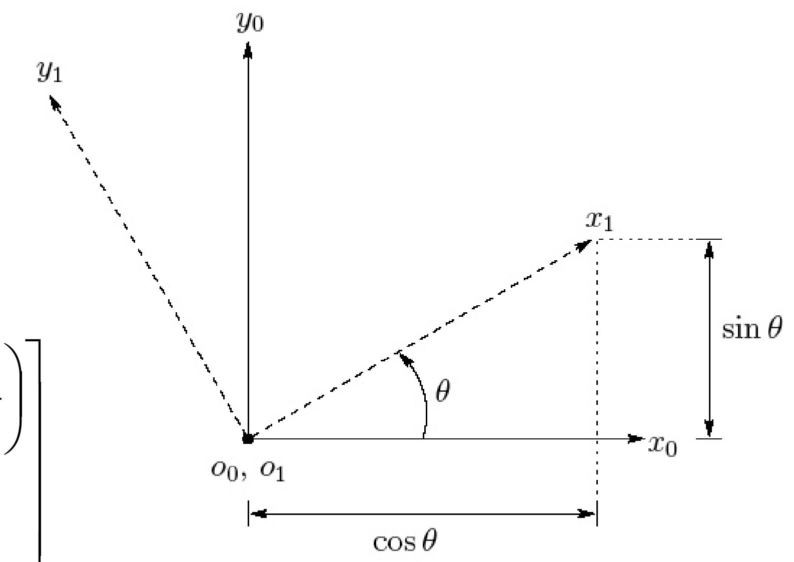
Projecting the axes of from o_1 onto the axes of frame o_0

$$x_1^0 = \begin{bmatrix} \hat{x}_1 \cdot \hat{x}_0 \\ \hat{x}_1 \cdot \hat{y}_0 \end{bmatrix}, y_1^0 = \begin{bmatrix} \hat{y}_1 \cdot \hat{x}_0 \\ \hat{y}_1 \cdot \hat{y}_0 \end{bmatrix}$$

$$R_1^0 = \begin{bmatrix} \hat{x}_1 \cdot \hat{x}_0 & \hat{y}_1 \cdot \hat{x}_0 \\ \hat{x}_1 \cdot \hat{y}_0 & \hat{y}_1 \cdot \hat{y}_0 \end{bmatrix}$$

$$= \begin{bmatrix} \|\hat{x}_1\|\|\hat{x}_0\|\cos\theta & \|\hat{y}_1\|\|\hat{x}_0\|\cos\left(\theta + \frac{\pi}{2}\right) \\ \|\hat{x}_1\|\|\hat{y}_0\|\cos\left(\frac{\pi}{2} - \theta\right) & \|\hat{y}_1\|\|\hat{y}_0\|\cos\theta \end{bmatrix}$$

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



Properties of rotation matrices

Summary:

Columns (rows) of R are mutually orthogonal

Each column (row) of R is a unit vector

$$R^T = R^{-1}$$
$$\det(R) = 1$$

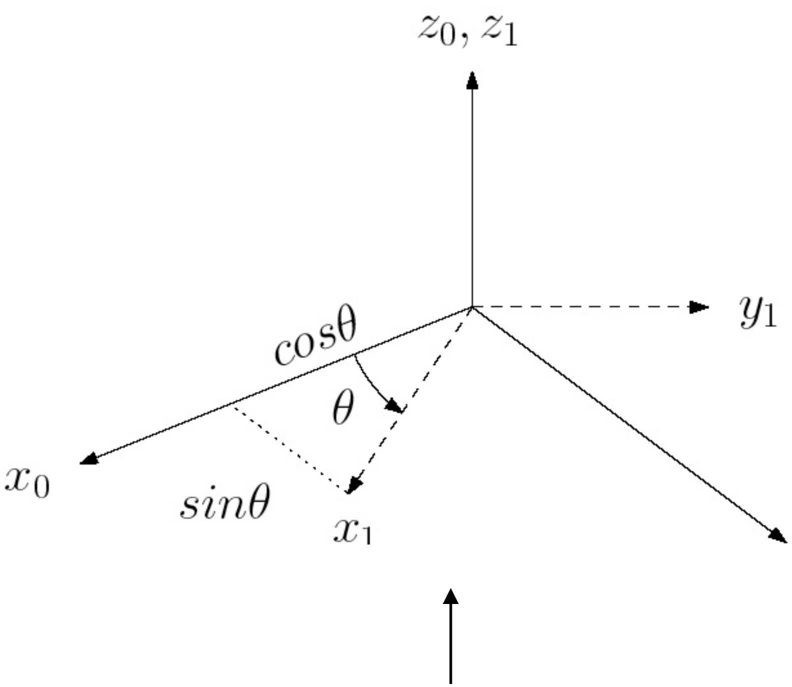
The set of all n x n matrices that have these properties are called the **Special Orthogonal group** of order n

$$R \in SO(n)$$

3D rotations

General 3D rotation:

$$R_1^0 = \begin{bmatrix} \hat{x}_1 \cdot \hat{x}_0 & \hat{y}_1 \cdot \hat{x}_0 & \hat{z}_1 \cdot \hat{x}_0 \\ \hat{x}_1 \cdot \hat{y}_0 & \hat{y}_1 \cdot \hat{y}_0 & \hat{z}_1 \cdot \hat{y}_0 \\ \hat{x}_1 \cdot \hat{z}_0 & \hat{y}_1 \cdot \hat{z}_0 & \hat{z}_1 \cdot \hat{z}_0 \end{bmatrix} \in SO(3)$$



Special cases

Basic rotation matrices

$$R_{x,\theta} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{bmatrix}$$

$$R_{y,\theta} = \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix}$$

$$R_{z,\theta} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Properties of rotation matrices (cont'd)

$SO(3)$ is a group under multiplication

Closure: **if** $R_1, R_2 \in SO(3) \Rightarrow R_1 R_2 \in SO(3)$

Identity: $I = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \in SO(3)$

Inverse: $R^T = R^{-1}$

Associativity: $(R_1 R_2) R_3 = R_1 (R_2 R_3)$ → Allows us to combine rotations: $R_{ac} = R_{ab} R_{bc}$

In general, members of $SO(3)$ do not commute

$$R_1 R_2 \neq R_2 R_1$$

Rotating a vector

Another interpretation of a rotation matrix:

Rotating a vector about an axis in a fixed frame

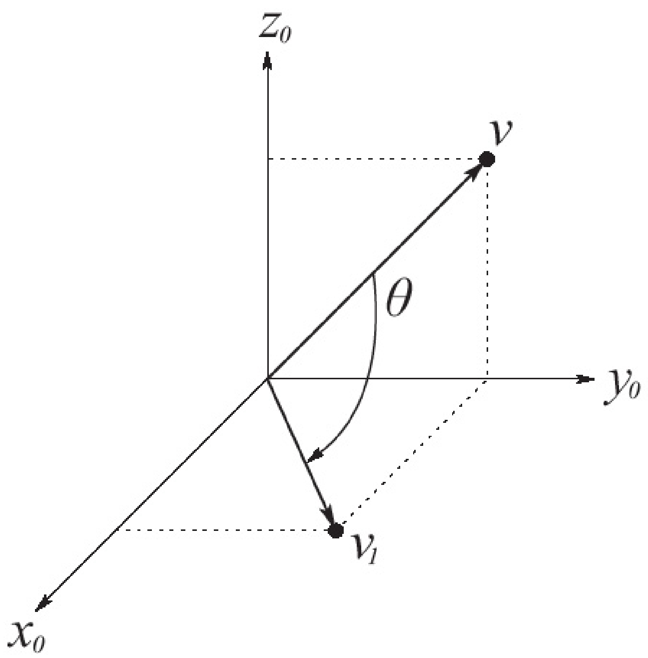
Ex: rotate v^0 about y_0 by $\pi/2$

$$v^0 = \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix}$$

$$v^1 = R_{y,\pi/2} v^0$$

$$= \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix}\Bigg|_{\theta=\pi/2} \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix}$$

$$= \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ -1 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}$$



Rotation matrix summary

Three interpretations for the role of rotation matrix:

- Representing the coordinates of a point in two different frames
- Orientation of a transformed coordinate frame with respect to a fixed frame
- Rotating vectors in the same coordinate frame

Compositions of rotations

w/ respect to the current frame

Ex: three frames o_0, o_1, o_2

$$\left.\begin{aligned} p^0 &= R_1^0 p^1 \\ p^1 &= R_2^1 p^2 \\ p^0 &= R_2^0 p^2 \end{aligned}\right\} \; p^0 = R_1^0 R_2^1 p^2 \longrightarrow R_2^0 = R_1^0 R_2^1$$

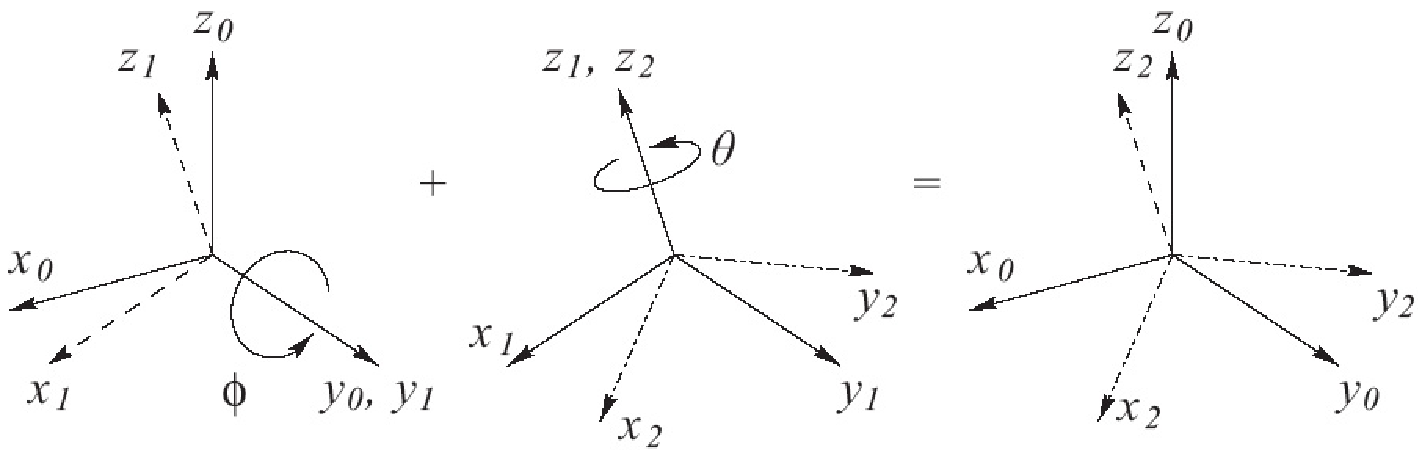
This defines the composition law for successive rotations about the **current** reference frame: post-multiplication

Compositions of rotations

Ex: R represents rotation about the current y-axis by ϕ followed by θ about the current z-axis

$$R = R_{y,\phi} R_{z,\theta}$$

$$= \begin{bmatrix} \cos\phi & 0 & \sin\phi \\ 0 & 1 & 0 \\ -\sin\phi & 0 & \cos\phi \end{bmatrix} \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \cos\phi\cos\theta & -\cos\phi\sin\theta & \sin\phi \\ \sin\theta & \cos\theta & 0 \\ -\sin\phi\cos\theta & \sin\phi\sin\theta & \cos\phi \end{bmatrix}$$



Compositions of rotations

w/ respect to a fixed reference frame (o_0)

Let the rotation between two frames o_0 and o_1 be defined by R_1^0

Let R be a desired rotation w/ respect to the fixed frame o_0

Using the definition of a similarity transform, we have:

$$R_2^0 = R_1^0 \left[\left(R_1^0 \right)^{-1} R R_1^0 \right] = R R_1^0$$

This defines the composition law for successive rotations about a **fixed** reference frame: pre-multiplication

Compositions of rotations

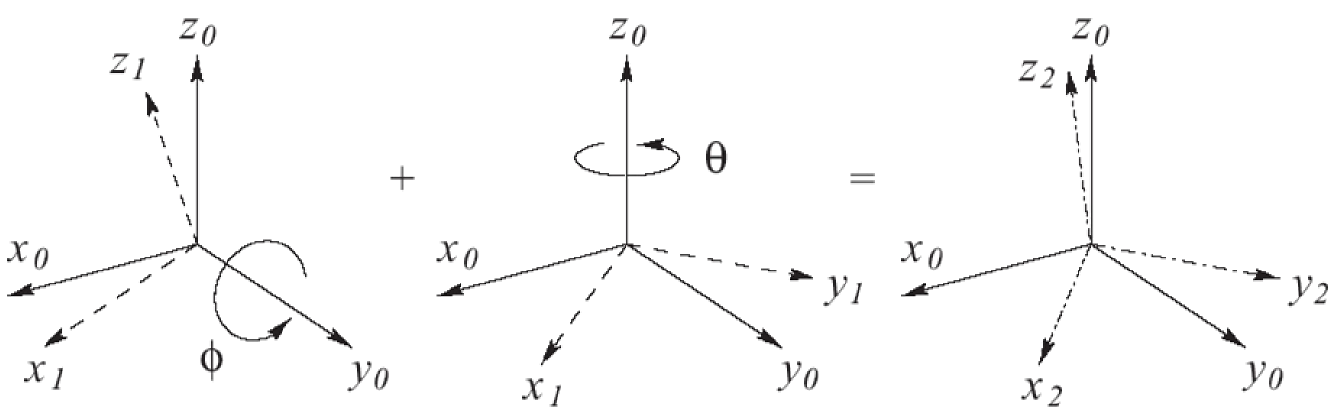
Ex: we want a rotation matrix R that is a composition of ϕ about y_0 ($R_{y,\phi}$) and then θ about z_0 ($R_{z,\theta}$)

the second rotation needs to be projected back to the initial fixed frame

$$R_2^0 = \left(R_{y,\theta}\right)^{-1} R_{z,\theta} R_{y,\theta}$$
$$= R_{y,-\theta} R_{z,\theta} R_{y,\theta}$$

Now the combination of the two rotations is:

$$R = R_{y,\phi}\left[R_{y,-\phi} R_{z,\theta} R_{y,\phi}\right] = R_{z,\theta} R_{y,\phi}$$



Compositions of rotations

Summary:

- Consecutive rotations w/ respect to the current reference frame:
  - Post-multiplying by successive rotation matrices
- w/ respect to a fixed reference frame (o_0)
  - Pre-multiplying by successive rotation matrices
- We can also have hybrid compositions of rotations with respect to the current and a fixed frame using these same rules

Parameterizing rotations

There are three parameters that need to be specified to create arbitrary rigid body rotations

We will describe three such parameterizations:

Euler angles

Roll, Pitch, Yaw angles

Axis/Angle

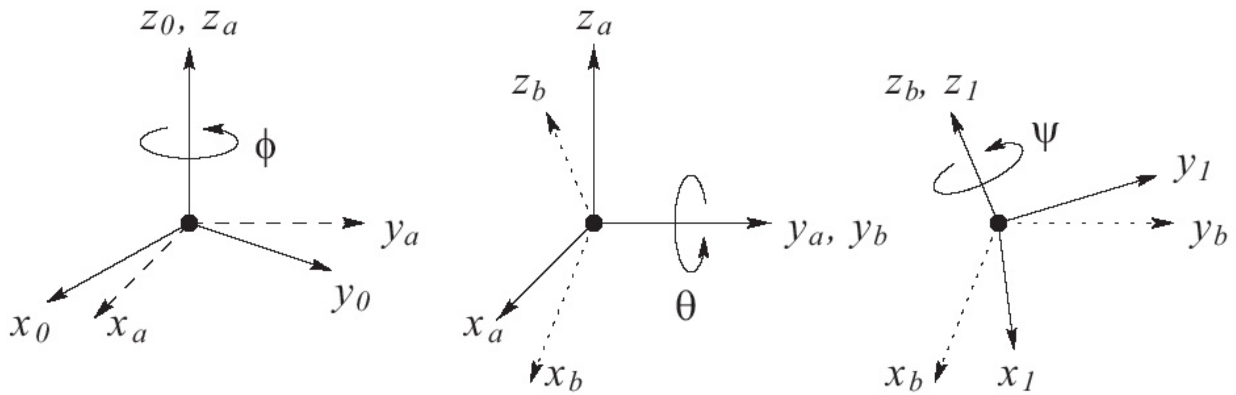
Parameterizing rotations

Euler angles

Rotation by ϕ about the z-axis, followed by θ about the current y-axis, then ψ about the current z-axis

$$R_{ZYZ} = R_{z,\phi}R_{y,\theta}R_{z,\psi} = \begin{bmatrix} c_\phi & -s_\phi & 0 \\ s_\phi & c_\phi & 0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} c_\theta & 0 & s_\theta \\ 0 & 1 & 0 \\ -s_\theta & 0 & c_\theta \end{bmatrix}\begin{bmatrix} c_\psi & -s_\psi & 0 \\ s_\psi & c_\psi & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} c_\phi c_\theta c_\psi - s_\phi s_\psi & -c_\phi c_\theta s_\psi - s_\phi c_\psi & c_\phi s_\theta \\ s_\phi c_\theta c_\psi + c_\phi s_\psi & -s_\phi c_\theta s_\psi + c_\phi c_\psi & s_\phi s_\theta \\ -s_\theta c_\psi & s_\theta s_\psi & c_\theta \end{bmatrix}$$

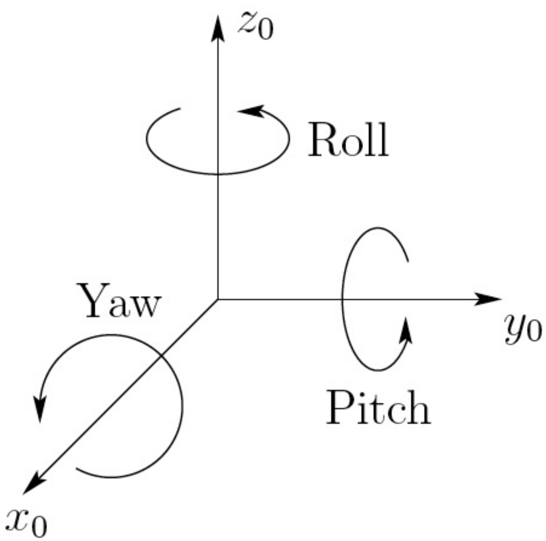


Parameterizing rotations

Roll, Pitch, Yaw angles

Three consecutive rotations about the fixed principal axes:

Yaw (x_0) ψ, pitch (y_0) θ, roll (z_0) ϕ



$$R_{XYZ} = R_{z,\phi} R_{y,\theta} R_{x,\psi}$$

$$= \begin{bmatrix} c_\phi & -s_\phi & 0 \\ s_\phi & c_\phi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c_\theta & 0 & s_\theta \\ 0 & 1 & 0 \\ -s_\theta & 0 & c_\theta \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & c_\psi & -s_\psi \\ 0 & s_\psi & c_\psi \end{bmatrix}$$

$$= \begin{bmatrix} c_\phi c_\theta & -s_\phi c_\psi + c_\phi s_\theta s_\psi & s_\phi s_\psi + c_\phi s_\theta c_\psi \\ s_\phi c_\theta & c_\phi c_\psi + s_\phi s_\theta s_\psi & -c_\phi s_\psi + s_\phi s_\theta c_\psi \\ -s_\theta & c_\theta s_\psi & c_\theta c_\psi \end{bmatrix}$$

Parameterizing rotations

Axis/Angle representation

Any rotation matrix in SO(3) can be represented as a single rotation about a suitable axis through a set angle

For example, assume that we have a unit vector:

$$\hat{k} = \begin{bmatrix} k_x \\ k_y \\ k_z \end{bmatrix}$$

Given θ, we want to derive $R_{k,\theta}$:

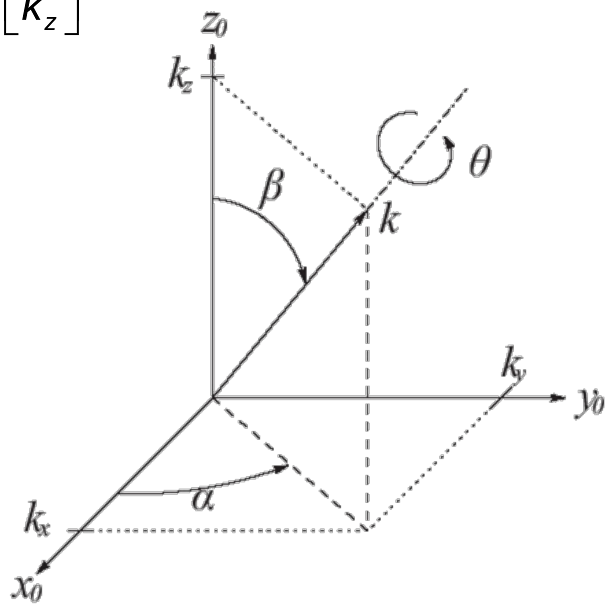
Intermediate step: project the *z*-axis onto *k*:

$$R_{k,\theta} = R R_{z,\theta} R^{-1}$$

Where the rotation *R* is given by:

$$R = R_{z,\alpha} R_{y,\beta}$$

$$\Rightarrow R_{k,\theta} = R_{z,\alpha} R_{y,\beta} R_{z,\theta} R_{y,-\beta} R_{z,-\alpha}$$



Parameterizing rotations

Axis/Angle representation

This is given by:

$$R_{k,\theta} = \begin{bmatrix} k_x^2 v_\theta + c_\theta & k_x k_y v_\theta - k_z s_\theta & k_x k_z v_\theta + k_y s_\theta \\ k_x k_y v_\theta + k_z s_\theta & k_y^2 v_\theta + c_\theta & k_y k_z v_\theta - k_x s_\theta \\ k_x k_z v_\theta - k_y s_\theta & k_y k_z v_\theta + k_x s_\theta & k_z^2 v_\theta + c_\theta \end{bmatrix}$$

Inverse problem:

Given arbitrary *R*, find *k* and θ

$$\theta = \cos^{-1}\left(\frac{Tr(R)-1}{2}\right)$$

$$\hat{k} = \frac{1}{2\sin\theta}\begin{bmatrix} r_{32} - r_{23} \\ r_{13} - r_{31} \\ r_{21} - r_{12} \end{bmatrix}$$

Rigid motions

Rigid motion is a combination of rotation and translation

Defined by a rotation matrix (R) and a displacement vector (d)

$$R \in SO(3)$$
$$d \in \mathbf{R}^3$$

the group of all rigid motions (d,R) is known as the **Special Euclidean group**, $SE(3)$

$$SE(3) = \mathbf{R}^3 \times SO(3)$$

Consider three frames, o_0, o_1, and o_2 and corresponding rotation matrices R_2^1, and R_1^0

Let d_2^1 be the vector from the origin o_1 to o_2, d_1^0 from o_0 to o_1

For a point p^2 attached to o_2, we can represent this vector in frames o_0 and o_1:

$$p^1 = R_2^1 p^2 + d_2^1$$
$$\begin{aligned} p^0 &= R_1^0 p^1 + d_1^0 \\ &= R_1^0 \left(R_2^1 p^2 + d_2^1\right) + d_1^0 \\ &= R_1^0 R_2^1 p^2 + R_1^0 d_2^1 + d_1^0 \end{aligned}$$

Homogeneous transforms

We can represent rigid motions (rotations and translations) as matrix multiplication

Define:

$$H_1^0 = \begin{bmatrix} R_1^0 & d_1^0 \\ 0 & 1 \end{bmatrix}$$

$$H_2^1 = \begin{bmatrix} R_2^1 & d_2^1 \\ 0 & 1 \end{bmatrix}$$

Now the point p_2 can be represented in frame o_0:

$$P^0 = H_1^0 H_2^1 P^2$$

Where the P^0 and P^2 are:

$$P^0 = \begin{bmatrix} p^0 \\ 1 \end{bmatrix}, P^2 = \begin{bmatrix} p^2 \\ 1 \end{bmatrix}$$

Homogeneous transforms

The matrix multiplication *H* is known as a **homogeneous transform** and we note that

$$H \in SE(3)$$

Inverse transforms:

$$H^{-1} = \begin{bmatrix} R^T & -R^T d \\ 0 & 1 \end{bmatrix}$$

Homogeneous transforms

Basic transforms:

Three pure translation, three pure rotation

$$\mathbf{Trans}_{x,a} = \begin{bmatrix} 1 & 0 & 0 & a \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad \mathbf{Rot}_{x,\alpha} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c_\alpha & -s_\alpha & 0 \\ 0 & s_\alpha & c_\alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{Trans}_{y,b} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & b \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad \mathbf{Rot}_{y,\beta} = \begin{bmatrix} c_\beta & 0 & s_\beta & 0 \\ 0 & 1 & 0 & 0 \\ -s_\beta & 0 & c_\beta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{Trans}_{z,c} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & c \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad \mathbf{Rot}_{z,\gamma} = \begin{bmatrix} c_\gamma & -s_\gamma & 0 & 0 \\ s_\gamma & c_\gamma & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Ch. 3: Forward and Inverse Kinematics

Recap: rigid motions

Rigid motion is a combination of rotation and translation

- Defined by a rotation matrix (*R*) and a displacement vector (*d*)
- the group of all rigid motions (*d*,*R*) is known as the **Special Euclidean group**, *SE*(3)

We can represent rigid motions (rotations and translations) as matrix multiplication

The matrix multiplication *H* is known as a **homogeneous transform** and we note that

$$H = \begin{bmatrix} R & d \\ 0 & 1 \end{bmatrix}$$

Inverse transforms:

$$H^{-1} = \begin{bmatrix} R^T & -R^T d \\ 0 & 1 \end{bmatrix}$$

Recap: homogeneous transforms

Basic transforms:

Three pure translation, three pure rotation

$$\mathbf{Trans}_{x,a} = \begin{bmatrix} 1 & 0 & 0 & a \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad \mathbf{Rot}_{x,\alpha} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c_\alpha & -s_\alpha & 0 \\ 0 & s_\alpha & c_\alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{Trans}_{y,b} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & b \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad \mathbf{Rot}_{y,\beta} = \begin{bmatrix} c_\beta & 0 & s_\beta & 0 \\ 0 & 1 & 0 & 0 \\ -s_\beta & 0 & c_\beta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{Trans}_{z,c} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & c \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad \mathbf{Rot}_{z,\gamma} = \begin{bmatrix} c_\gamma & -s_\gamma & 0 & 0 \\ s_\gamma & c_\gamma & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Example

Euler angles: we have only discussed ZYZ Euler angles. What is the set of all possible Euler angles that can be used to represent any rotation matrix?

Answer - Euler

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

ZZY cannot be used to describe any arbitrary rotation matrix since two consecutive rotations about the Z axis can be composed into one rotation

Example

Compute the homogeneous transformation representing a translation of 3 units along the x-axis followed by a rotation of $\pi/2$ about the current z-axis followed by a translation of 1 unit along the fixed y-axis

Answer – Homogeneous Transforms

$$T = T_{y,1}T_{x,3}T_{z,\pi/2}$$

$$= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 3 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} 0 & -1 & 0 & 3 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Forward kinematics introduction

Challenge: given all the joint parameters of a manipulator, determine the position and orientation of the tool frame

- Tool frame: coordinate frame attached to the most distal link of the manipulator
- Inertial (base) frame: fixed (immobile) coordinate system fixed to the most proximal link of a manipulator

Therefore, we want a mapping between the tool frame and the inertial frame

- This will be a function of all joint parameters and the physical geometry of the manipulator

Purely geometric: we do not worry about joint torques or dynamics

- (yet!)

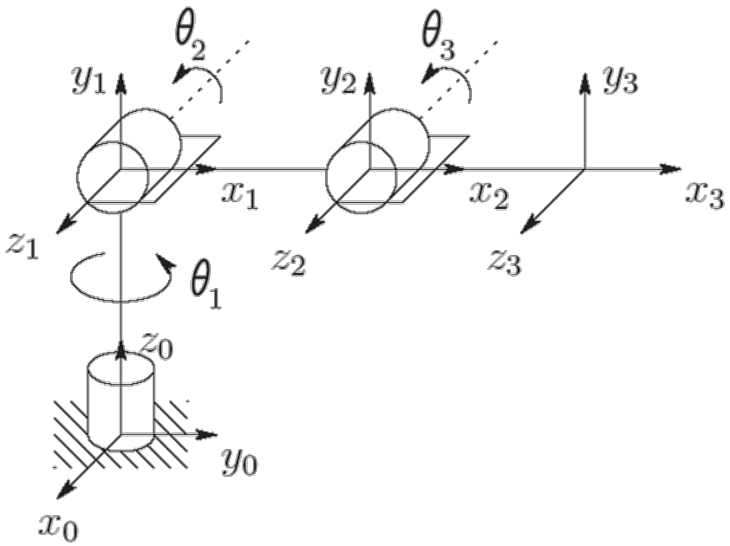
Convention

A n-DOF manipulator will have n joints (either revolute or prismatic) and n+1 links (since each joint connects two links)

- We assume that each joint only has one DOF. Although this may seem like it does not include things like spherical or universal joints, we can think of multi-DOF joints as a combination of 1DOF joints with zero length between them
- The o_0 frame is the inertial frame (or base frame)
- o_n is the tool frame
- Joint i connects links i-1 and i
- The o_i is connected to link i

Joint variables, q_i

$$q_i = \begin{cases} \theta_i & \textbf{if joint } i \textbf{ is revolute} \\ d_i & \textbf{if joint } i \textbf{ is prismatic} \end{cases}$$



Convention

We said that a homogeneous transformation allowed us to express the position and orientation of o_j with respect to o_i

what we want is the position and orientation of the tool frame with respect to the inertial frame

An intermediate step is to determine the transformation matrix that gives position and orientation of o_i with respect to o_{i-1}: A_i

Now we can define the transformation o_j to o_i as:

$$T_j^i = \begin{cases} A_{i+1}A_{i+2}\ldots A_{j-i}A_j & \textbf{if } i < j \\ I & \textbf{if } i = j \\ \left(T_i^j\right)^{-1} & \textbf{if } j > i \end{cases}$$

Convention

Finally, the position and orientation of the tool frame with respect to the inertial frame is given by one homogeneous transformation matrix:

For a n-DOF manipulator

$$H = \begin{bmatrix} R_n^0 & o_n^0 \\ 0 & 1 \end{bmatrix} = T_n^0 = A_1(q_1)A_2(q_2)\cdots A_n(q_n)$$

Thus, to fully define the forward kinematics for any serial manipulator, all we need to do is create the A_i transformations and perform matrix multiplication

But there are shortcuts…

The Denavit-Hartenberg (DH) Convention

Representing each individual homogeneous transformation as the product of four basic transformations:

$$A_i = \mathbf{Rot}_{z,\theta_i}\mathbf{Trans}_{z,d_i}\mathbf{Trans}_{x,a_i}\mathbf{Rot}_{x,\alpha_i}$$

$$= \begin{bmatrix} c_{\theta_i} & -s_{\theta_i} & 0 & 0 \\ s_{\theta_i} & c_{\theta_i} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & a_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c_{\alpha_i} & -s_{\alpha_i} & 0 \\ 0 & s_{\alpha_i} & c_{\alpha_i} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} c_{\theta_i} & -s_{\theta_i}c_{\alpha_i} & s_{\theta_i}s_{\alpha_i} & a_i c_{\theta_i} \\ s_{\theta_i} & c_{\theta_i}c_{\alpha_i} & -c_{\theta_i}s_{\alpha_i} & a_i s_{\theta_i} \\ 0 & s_{\alpha_i} & c_{\alpha_i} & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The Denavit-Hartenberg (DH) Convention

Four DH parameters:

- a_i: link length
- α_i: link twist
- d_i: link offset
- θ_i: joint angle

Since each A_i is a function of only one variable, three of these will be constant for each link

- d_i will be variable for prismatic joints and θ_i will be variable for revolute joints

But we said any rigid body needs 6 parameters to describe its position and orientation

- Three angles (Euler angles, for example) and a 3x1 position vector
- So how can there be just 4 DH parameters?...

Existence and uniqueness

When can we represent a homogeneous transformation using the 4 DH parameters?

For example, consider two coordinate frames o_0 and o_1

There is a unique homogeneous transformation between these two frames

Now assume that the following holds:

DH1: perpendicular -> $\hat{x}_1 \perp \hat{z}_0$

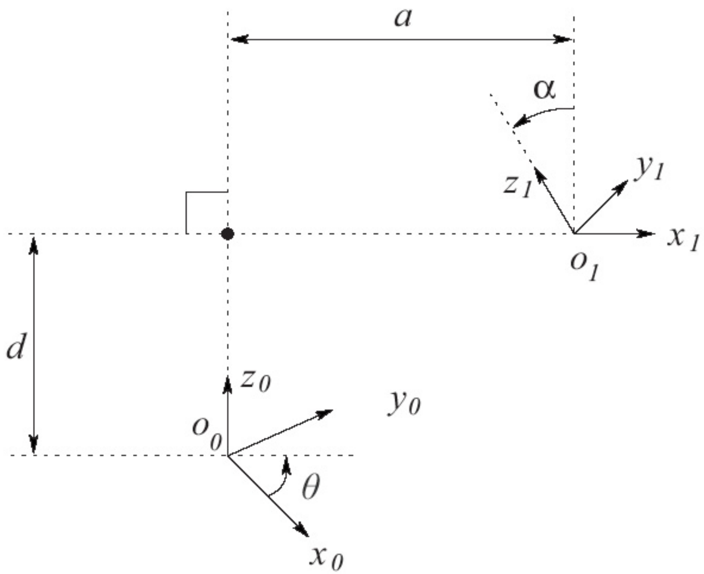
DH2: intersects -> $\hat{x}_1 \cap \hat{z}_0$

If these hold, we claim that there

exists a unique transformation A:

$$A = \mathbf{Rot}_{z,\theta}\mathbf{Trans}_{z,d}\mathbf{Trans}_{x,a}\mathbf{Rot}_{x,\alpha}$$

$$= \begin{bmatrix} R_1^0 & o_1^0 \\ 0 & 1 \end{bmatrix}$$



Existence and uniqueness

Proof:

We assume that R_1^0 has the form: $R_1^0 = R_{z,\theta} R_{x,\alpha}$

Use DH1 to verify the form of R_1^0

$$\hat{x}_1 \perp \hat{z}_0 \Rightarrow x_1^0 \cdot z_0^0 = 0$$

$$\Rightarrow \begin{bmatrix} r_{11} \\ r_{21} \\ r_{31} \end{bmatrix}^T \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = r_{31} = 0 \longrightarrow R_1^0 = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ 0 & r_{32} & r_{33} \end{bmatrix}$$

Since the rows and columns of R_1^0 must be unit vectors:

$$\longrightarrow \begin{aligned} r_{11}^2 + r_{21}^2 &= 1 \\ r_{32}^2 + r_{33}^2 &= 1 \end{aligned}$$

The remainder of R_1^0 follows from the properties of rotation matrices

Therefore our assumption that there exists a unique θ and α that will give us R_1^0 is correct given DH1

Existence and uniqueness

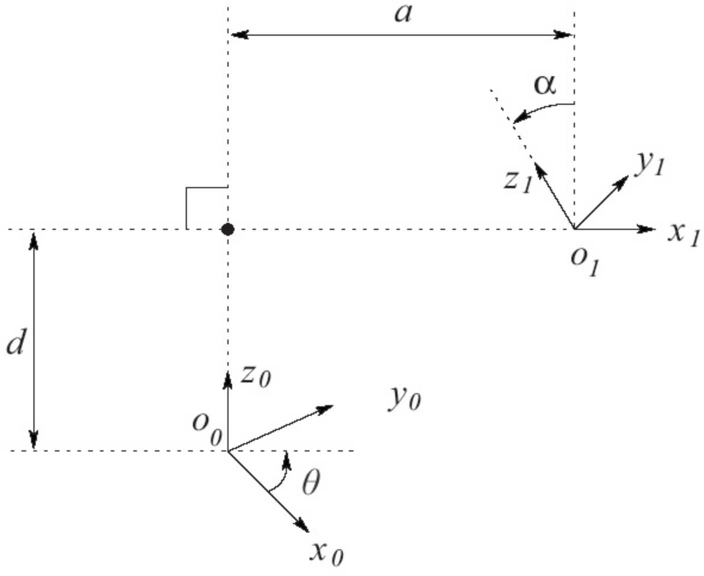
Proof:

Use DH2 to determine the form of o_1^0

Since the two axes intersect, we can represent the line between the two frames as a linear combination of the two axes (within the plane formed by x_1 and z_0)

$$\hat{x}_1 \cap \hat{z}_0 \Rightarrow o_1^0 = dz_0^0 + ax_1^0$$

$$\Rightarrow o_1^0 = d\begin{bmatrix}0\\0\\1\end{bmatrix} + a\begin{bmatrix}c_\theta\\s_\theta\\0\end{bmatrix} = \begin{bmatrix}ac_\theta\\as_\theta\\d\end{bmatrix}$$



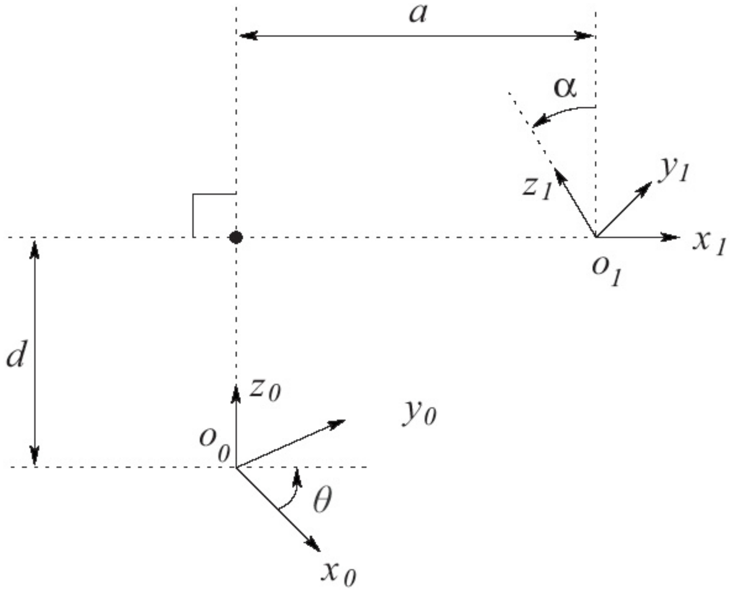
Physical basis for DH parameters

a_i: link length, distance between the z_0 and z_1 (along x_1)

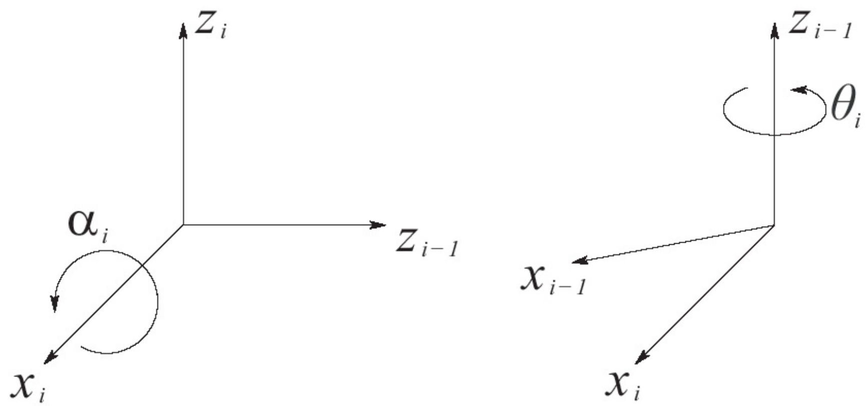
α_i: link twist, angle between z_0 and z_1 (measured around x_1)

d_i: link offset, distance between o_0 and intersection of z_0 and x_1 (along z_0)

θ_i: joint angle, angle between x_0 and x_1 (measured around z_0)



positive convention:



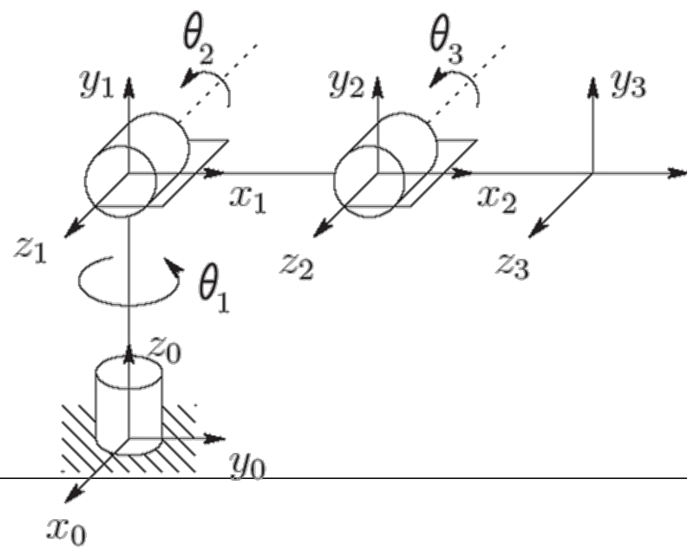
Assigning coordinate frames

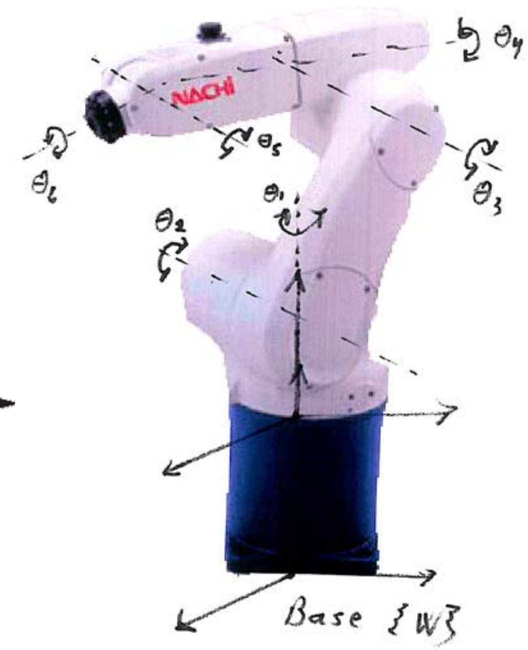
For any *n*-link manipulator, we can always choose coordinate frames such that DH1 and DH2 are satisfied

- The choice is not unique, but the end result will always be the same

Choose z_i as axis of rotation for joint *i*+1

- z_0 is axis of rotation for joint 1, z_1 is axis of rotation for joint 2, etc
- If joint *i*+1 is revolute, z_i is the axis of rotation of joint *i*+1
- If joint *i*+1 is prismatic, z_i is the axis of translation for joint *i*+1





Assigning coordinate frames

Assign base frame

Can be any point along z_0

Chose x_0, y_0 to follow the right-handed convention

Now start an iterative process to define frame *i* with respect to frame *i*-1

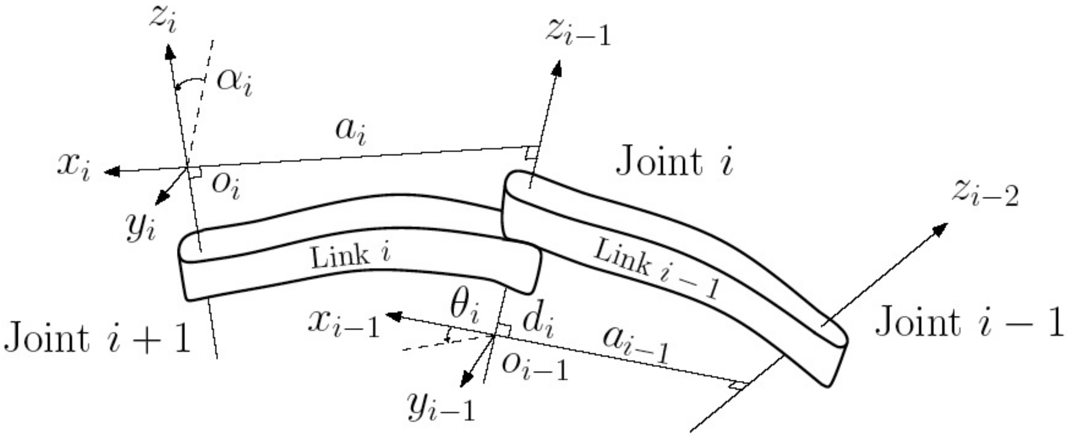
Consider three cases for the relationship of z_{i-1} and z_i:

z_{i-1} and z_i are non-coplanar

z_{i-1} and z_i intersect

z_{i-1} and z_i are parallel

} z_{i-1} and z_i are coplanar



Assigning coordinate frames

$z_{i\text{-}1}$ and z_i are non-coplanar

- There is a unique shortest distance between the two axes
- Choose this line segment to be x_i
  - o_i is at the intersection of z_i and x_i
- Choose y_i by right-handed convention

Assigning coordinate frames

z_{i-1} and z_i intersect

- Choose x_i to be normal to the plane defined by z_i and z_{i-1}
  - o_i is at the intersection of z_i and x_i
- Choose y_i by right-handed convention

Assigning coordinate frames

z_{i-1} and z_i are parallel

- Infinitely many normals of equal length between z_i and z_{i-1}
- Free to choose o_i anywhere along z_i, however if we choose x_i to be along the normal that intersects at o_{i-1}, the resulting d_i will be zero
- Choose y_i by right-handed convention

Assigning tool frame

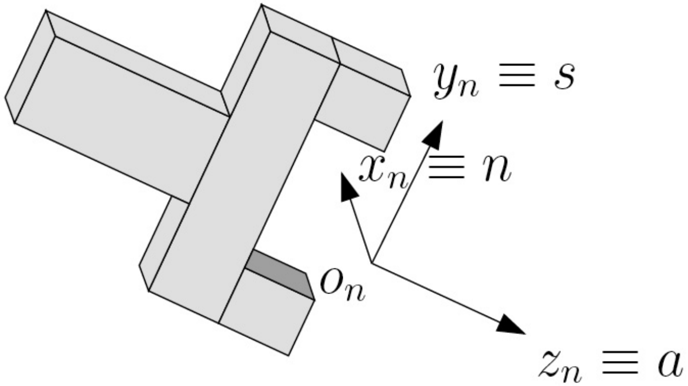
The previous assignments are valid up to frame *n*-1

The tool frame assignment is most often defined by the axes *n*, *s*, *a*:

a is the approach direction

s is the 'sliding' direction (direction along which the grippers open/close)

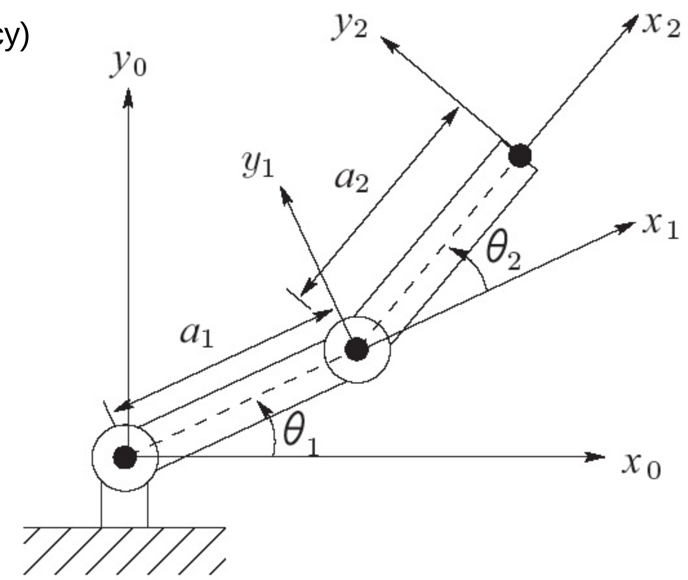
n is the normal direction to *a* and *s*



Example 1: two-link planar manipulator

2DOF: need to assign three coordinate frames

- Choose z_0 axis (axis of rotation for joint 1, base frame)
- Choose z_1 axis (axis of rotation for joint 2)
- Choose z_2 axis (tool frame)
  - This is arbitrary for this case since we have described no wrist/gripper
  - Instead, define z_2 as parallel to z_1 and z_0 (for consistency)
- Choose x_i axes
  - All z_i's are parallel
  - Therefore choose x_i to intersect o_{i-1}



Example 1: two-link planar manipulator

Now define DH parameters

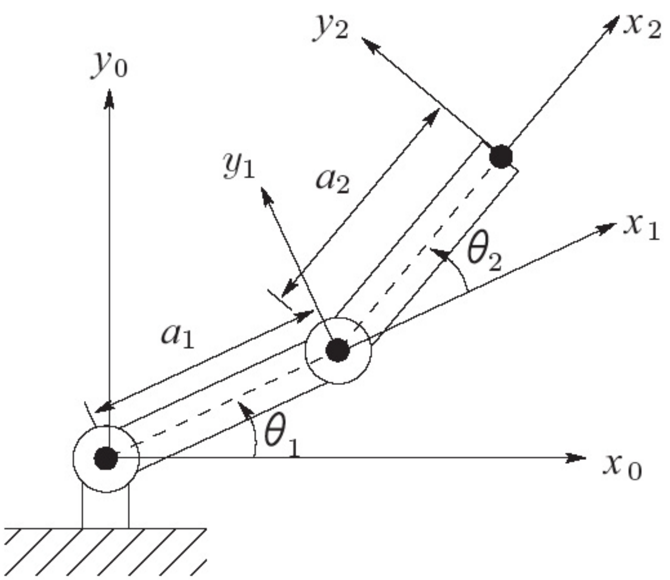
First, define the constant parameters a_i, α_i

Second, define the variable parameters θ_i, d_i

link	a_i	α_i	d_i	θ_i
1	a_1	0	0	θ_1
2	a_2	0	0	θ_2

The α_i terms are 0 because all z_i are parallel

Therefore only θ_i are variable



$$A_1 = \begin{bmatrix} c_1 & -s_1 & 0 & a_1c_1 \\ s_1 & c_1 & 0 & a_1s_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, A_2 = \begin{bmatrix} c_2 & -s_2 & 0 & a_2c_2 \\ s_2 & c_2 & 0 & a_2s_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

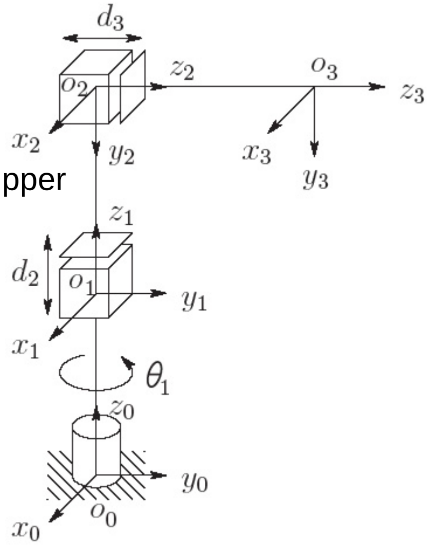
$$T_1^0 = A_1$$

$$T_2^0 = A_1A_2 = \begin{bmatrix} c_{12} & -s_{12} & 0 & a_1c_1 + a_2c_{12} \\ s_{12} & c_{12} & 0 & a_1s_1 + a_2s_{12} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Example 2: three-link cylindrical robot

3DOF: need to assign four coordinate frames

- Choose z_0 axis (axis of rotation for joint 1, base frame)
- Choose z_1 axis (axis of translation for joint 2)
- Choose z_2 axis (axis of translation for joint 3)
- Choose z_3 axis (tool frame)
  - This is again arbitrary for this case since we have described no wrist/gripper
  - Instead, define z_3 as parallel to z_2



Example 2: three-link cylindrical robot

Now define DH parameters

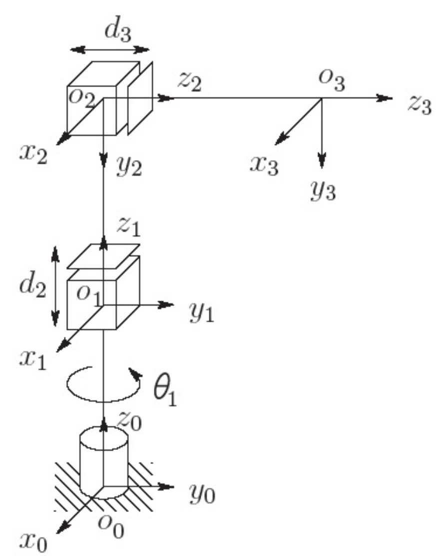
First, define the constant parameters a_i, α_i

Second, define the variable parameters θ_i, d_i

link	a_i	α_i	d_i	θ_i
1	0	0	d_1	θ_1
2	0	-90	d_2	0
3	0	0	d_3	0

$$A_1 = \begin{bmatrix} c_1 & -s_1 & 0 & 0 \\ s_1 & c_1 & 0 & 0 \\ 0 & 0 & 1 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}, A_2 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & d_2 \\ 0 & 0 & 0 & 1 \end{bmatrix}, A_3 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

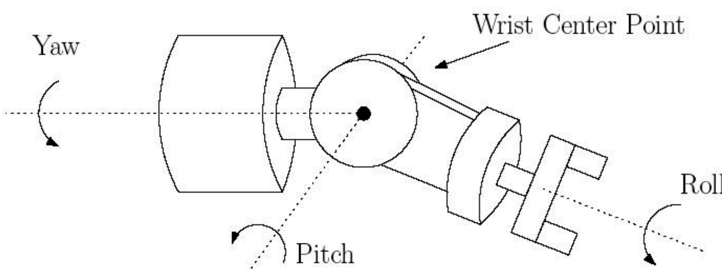
$$T_3^0 = A_1 A_2 A_3 = \begin{bmatrix} c_1 & 0 & -s_1 & -s_1 d_3 \\ s_1 & 0 & c_1 & c_1 d_3 \\ 0 & -1 & 0 & d_1 + d_2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

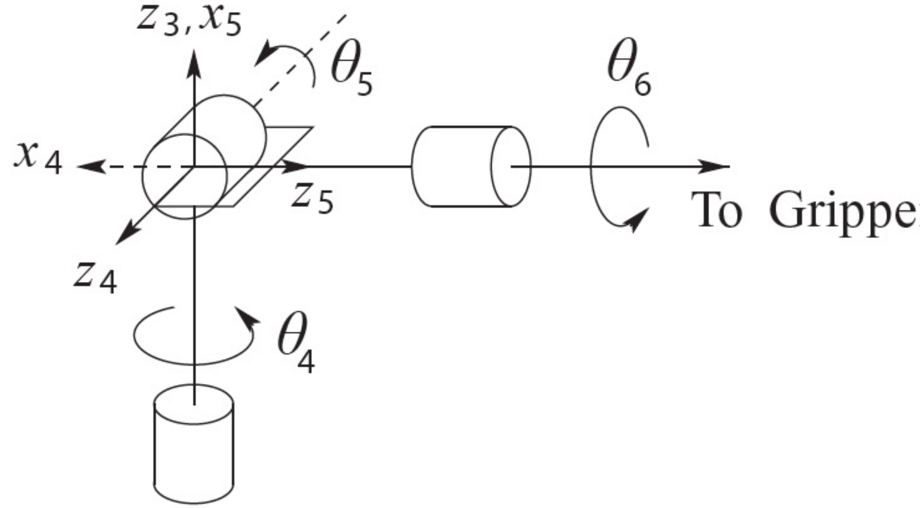


Example 3: spherical wrist

3DOF: need to assign four coordinate frames

yaw, pitch, roll (θ_4, θ_5, θ_6) all intersecting at one point *o* (wrist center)





Example 3: spherical wrist

Now define DH parameters

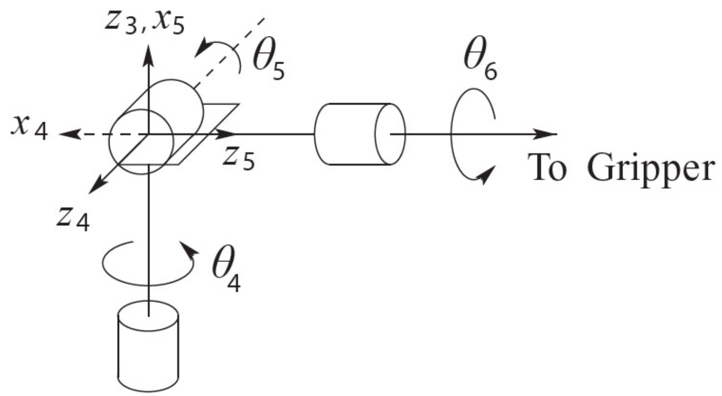
First, define the constant parameters a_i, α_i

Second, define the variable parameters θ_i, d_i

link	a_i	α_i	d_i	θ_i
4	0	-90	0	θ_4
5	0	90	0	θ_5
6	0	0	d_6	θ_6

$$A_1 = \begin{bmatrix} c_4 & 0 & -s_4 & 0 \\ s_4 & 0 & c_4 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, A_2 = \begin{bmatrix} c_5 & 0 & -s_5 & 0 \\ s_5 & 0 & c_5 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, A_3 = \begin{bmatrix} c_6 & -s_6 & 0 & 0 \\ s_6 & c_6 & 0 & 0 \\ 0 & 0 & 1 & d_6 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_6^3 = A_4 A_5 A_6 = \begin{bmatrix} c_4c_5c_6 - s_4s_6 & -c_4c_5s_6 - s_4c_6 & c_4s_5 & c_4s_5d_6 \\ s_4c_5c_6 + c_4s_6 & -s_4c_5s_6 + c_4c_6 & s_4s_5 & s_4s_5d_6 \\ -s_5c_6 & s_5c_6 & c_5 & c_5d_6 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



Next class...

More examples for common configurations

Link to movie that explains how to set-up the Denavit-Hartenberg parameters :

http://en.wikipedia.org/wiki/File:Denavit-Hartenberg_Tutorial_Video.ogv#file