§1. Important Basics

DoF : number of independent movements the robot is capable of

= # of joints for manipulator

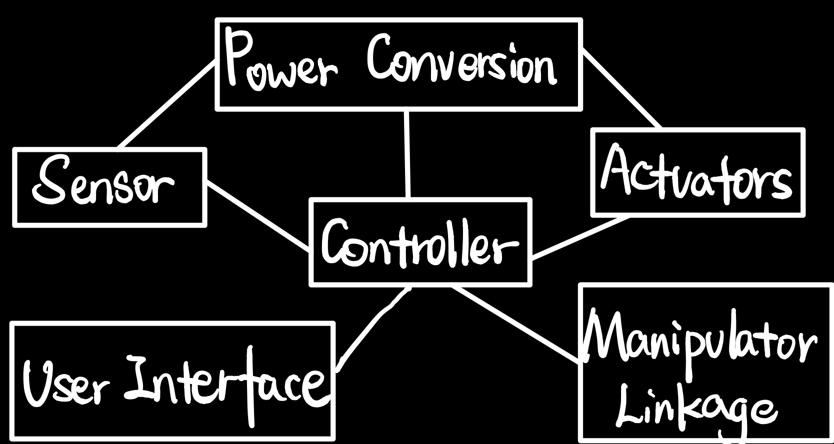
- Prismatic Joints : Linear motion along a straight line
- Revolute Joints : Rotational

Mechanical Types Configuration

- Cartesian : PPP >> Gantry Type *3 prismatic Joints
- Cylindrical : RPP *Rotation + 2 Prismat.
- Spherical : RRP >> SCARA
- Anthropomorphic : RRR *Most common

For Cables

*Workspace : Reachable space of robots (with orientation)



A Problem of R : Accumulated Error

§2. Manipulator Kinematics

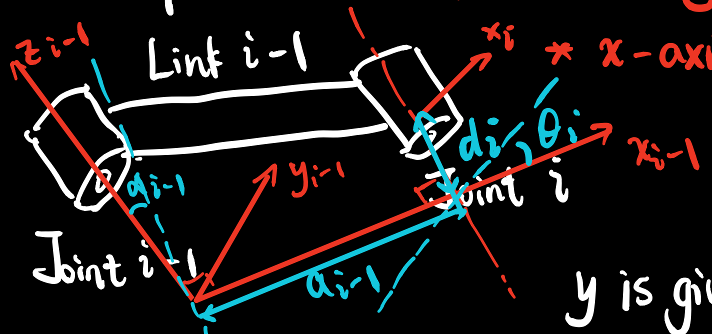
A set of joint variables θ. $\vec{x} = f(\vec{\theta})$ *Forward Kinematics

$\vec{\theta} = f^{-1}(\vec{x})$ *Inverse Kinematics

Rotation & Homogeneous Transformation Matrix

$$^{A}T_{B} = \begin{bmatrix} ^{A}R_{B} & \vec{p} \\ 0 & 1 \end{bmatrix}$$

- D-H Representation



* z-axis is given by motion direction

* x-axis given by perpendicular line of the 2 axis pointing to i

y is given by x & z

*Last frame's x parallel to the last one

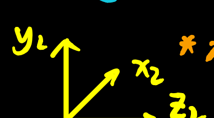
a_{i-1} : distance from z_{i-1} to z_i along x_{i-1}

α_{i-1} : angle from z_{i-1} to z_i about x_{i-1}

d_i : distance from x_{i-1} to x_i along z_i

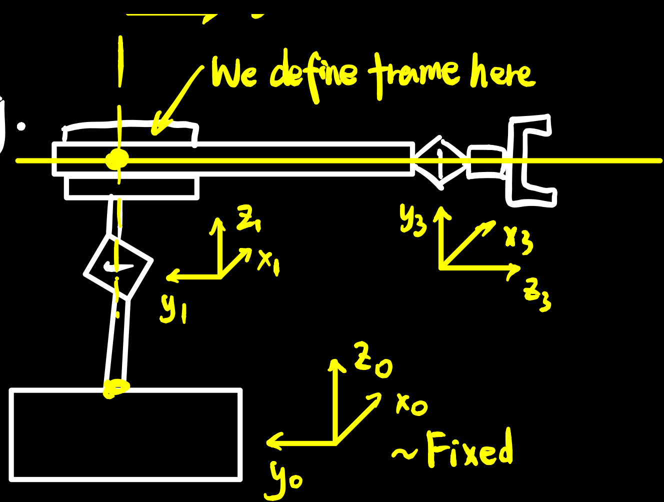
θ_i : angle from x_{i-1} to x_i about z_i

D-H Parameters



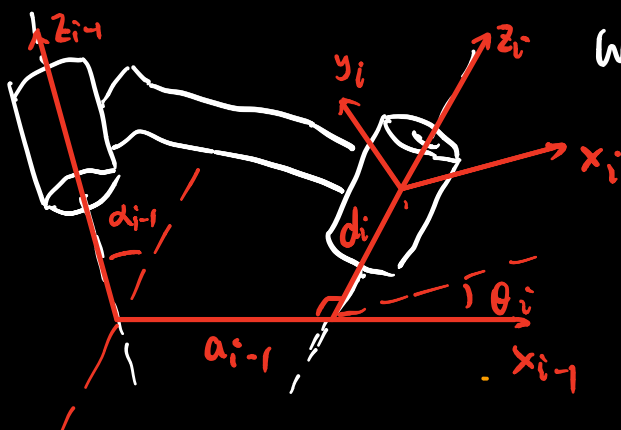
* x can be any where . for convenience here

e.g.



i	α_{i-1}	a_{i-1}	d_i	θ_i
1	0	0	0	θ_1
2	90°	0	d_2	0
3	0	0	0	θ_3

Transformation between Frames



we can make

$C_{i-1} = C_i$

By D-H parameters

$$
{}^{i-1}T_i = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & C\alpha_{i-1} & -S\alpha_{i-1} & 0 \\ 0 & S\alpha_{i-1} & C\alpha_{i-1} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} & I & & a_{i-1} \\ & & & 0 \\ & & & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} C\theta_i & -S\theta_i & 0 & 0 \\ S\theta_i & C\theta_i & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} I & \begin{matrix} 0 \\ 0 \\ d_i \end{matrix} \\ 0 & 1 \end{bmatrix}
$$

$$
= \begin{bmatrix} C\theta_i & -S\theta_i & 0 & a_{i-1} \\ S\theta_i C\alpha_{i-1} & C\theta_i C\alpha_{i-1} & -S\alpha_{i-1} & -S\alpha_{i-1} d_i \\ S\theta_i S\alpha_{i-1} & C\theta S\alpha_{i-1} & C\alpha_{i-1} & C\alpha_{i-1} d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

* Only 1 Variables here

• Inverse Kinematics

Given desired position & orientation

$$Y = (x, y, z, O, A, T)$$

$\Rightarrow$ How to get $q = (\theta_1, \theta_2 \cdots \theta_n)$

* Since we get ${}^0T_n = \begin{bmatrix} R_{11} & R_{12} & R_{13} & P_1 \\ R_{21} & R_{22} & R_{23} & P_2 \\ R_{31} & R_{32} & R_{33} & P_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$

~ we can get 12 equations by value of R & p

However it's highly NON-LINEAR!

$\Rightarrow$ { Dextrous Workspace : volume of space where the end effector can be arbitrarily oriented

Reachable Workspace : volume of space where robot reaches

α. General Solution

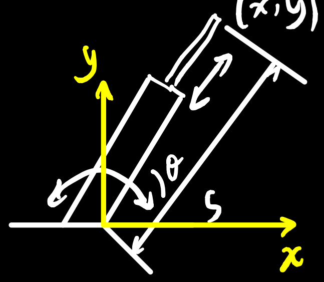
$$\underbrace{{}^0T_1^{-1} \cdot {}^0T_N}_{\text{Only 1 Variable lhs!}} = \underbrace{{}^1T_2 \, {}^2T_3 \cdots {}^{N-1}T_N}_{\text{N-1 Variables rhs.}}$$

* Some constant exists rhs

$\Rightarrow \theta_1$ can be solved

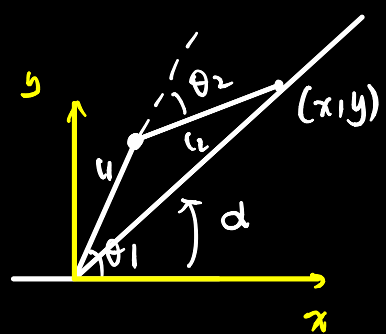
⚠ Not always work!

e.g.



$$\begin{cases} x = s\cos\theta \\ y = s\sin\theta \end{cases} \sim FK$$

$$\begin{cases} s = \sqrt{x^2+y^2} \\ \theta = \arctan\frac{y}{x} \end{cases} \sim IK$$



$$s^2 = l_1^2 + l_2^2 + 2l_1l_2\cos\theta_2$$

$$\begin{cases} x = s\cos\alpha = l_1\cos\theta_1 + l_2\cos(\theta_2-\theta_1) \\ y = s\sin\alpha = l_1\sin\theta_1 + l_1\sin(\theta_2-\theta_1) \end{cases} \sim FK$$

IK: $\dfrac{l_2}{\sin(\theta_1-\alpha)} = \dfrac{l_1}{\sin(\theta_2-\theta_1+\alpha)} = \dfrac{\sqrt{x^2+y^2}}{\sin\theta_2}$

$$x^2+y^2 = l_1^2 + l_2^2 + 2l_1l_2\cos\theta_2$$

$$\begin{cases} \theta_2 = \arccos\dfrac{x^2+y^2-l_1^2-l_2^2}{2l_1l_2} \\ \theta_1 = \arctan\dfrac{y}{x} + \arcsin\dfrac{l_2\sin\theta_2}{\sqrt{x^2+y^2}} \end{cases} \sim IK$$

* They can only used in simple structure

β. Numerical

$F(\theta) = [\underline{f(\theta)} - x]^2$, $\theta = \underset{x}{\arg\min}\, F(\theta)$

Hypothesized Position

Actual position

for-loops to search the whole θ state space

- Gauss-Newton Approach

Step 1. Initialize θ_0

Step 2. $\theta^{k+1} = \theta^k + \Delta\theta^k$

$$\underset{\Delta\theta^k}{\operatorname{argmin}} \sum_t \left[f_t(\theta^k + \Delta\theta^k) - c_t \right]^2$$

$$= \underset{\Delta\theta^k}{\operatorname{argmin}} \sum_t \left[f_t(\theta^k) + \frac{\partial f_t}{\partial \theta^k} \Delta\theta^k - c_t \right]^2$$ * Affine Inside

For optimal

$$J\Delta\theta = b \sim \begin{bmatrix} \frac{\partial f_1}{\partial \theta_1} & \cdots & \frac{\partial f_1}{\partial \theta_N} \\ \vdots & & \\ \frac{\partial f_N}{\partial \theta_1} & \cdots & \frac{\partial f_N}{\partial \theta_N} \end{bmatrix} \begin{bmatrix} \Delta\theta_1 \\ \Delta\theta_2 \\ \vdots \\ \Delta\theta_N \end{bmatrix} = \begin{bmatrix} c_1 - f_1(\theta_1 \cdots \theta_N) \\ c_2 - f_1(\theta_1 \cdots \theta_N) \\ \vdots \\ c_N - f_N(\theta_1 \cdots \theta_N) \end{bmatrix}$$