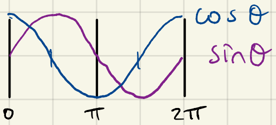
Forward Kinematics for EEzyBot Arm

27th Sept 2023
Ben Money-Coomes.

Method: Devanit Hartenberg | Proximal Method

N.B. The calculations presented here use the labelled parameters given in the Github repo easy EEZY bot ARM

Useful graphs →



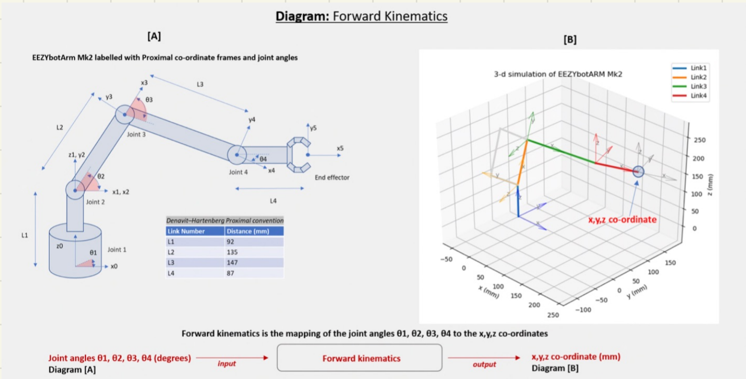
Notation → $c_1 c_2$ means $\cos\theta_1 \cos\theta_2$
c_{23} means $\cos(\theta_2+\theta_3)$

Useful trig identities

$$\sin^2\theta + \cos^2\theta = 1$$
$$\sin(\alpha \pm \beta) = \sin\alpha\cos\beta \pm \cos\alpha\sin\beta$$
$$\cos(\alpha \pm \beta) = \cos\alpha\cos\beta \mp \sin\alpha\sin\beta$$
$$\tan\theta = \frac{\sin\theta}{\cos\theta}$$

$R(x)\ T(x)\ R(z)\ T(z)$

n	a_{n-1}	α_{n-1}	d_n	θ_n
1	0	0	L_1	θ_1
2	0	$\pi/2$	0	θ_2
3	L_2	0	0	θ_3
4	L_3	0	0	θ_4



$$ {}^{n-1}_{n}T = R_{x_{n-1}}(\alpha_{n-1})\, T_{x_{n-1}}(a_{n-1})\, R_{z_n}(\theta_n)\, T_{z_n}(d_n) $$

$$ {}^{0}_{1}T = \begin{bmatrix} c_1 & -s_1 & 0 & 0 \\ s_1 & c_1 & 0 & 0 \\ 0 & 0 & 1 & L_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad {}^{1}_{2}T = \begin{bmatrix} c_2 & -s_2 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ s_2 & c_2 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad {}^{2}_{3}T = \begin{bmatrix} c_3 & -s_3 & 0 & L_2 \\ s_3 & c_3 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad {}^{3}_{4}T = \begin{bmatrix} c_4 & -s_4 & 0 & L_3 \\ s_4 & c_4 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

$$ {}^{0}_{2}T = \begin{bmatrix} c_1c_2 & -c_1s_2 & s_1 & 0 \\ s_1c_2 & -s_1s_2 & -c_1 & 0 \\ s_2 & c_2 & 0 & L_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

$$ {}^{0}_{3}T = {}^{0}_{2}T\, {}^{2}_{3}T = \begin{bmatrix} c_1c_2c_3 - c_1s_2s_3 & -c_1c_2s_3 - c_1s_2c_3 & s_1 & c_1c_2L_2 \\ s_1c_2c_3 - s_1s_2s_3 & -s_1c_2s_3 - s_1s_2c_3 & -c_1 & s_1c_2L_2 \\ s_2c_3 + c_2s_3 & -s_2s_3 + c_2c_3 & 0 & s_2L_2 + L_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} c_1c_{23} & -c_1s_{23} & s_1 & c_1c_2L_2 \\ s_1c_{23} & -s_1s_{23} & -c_1 & s_1c_2L_2 \\ s_{23} & c_{23} & 0 & s_2L_2 + L_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

$$ {}^{0}_{3}T\, {}^{3}_{4}T = {}^{0}_{4}T = \begin{bmatrix} c_1c_{23}c_4 - c_1s_{23}s_4 & -c_1c_{23}s_4 - c_1c_{23}s_4 & s_1 & c_1c_{23}L_3 + c_1c_2L_2 \\ s_1c_{23}c_4 - s_1s_{23}s_4 & -s_1c_{23}s_4 - s_1s_{23}c_4 & -c_1 & s_1c_{23}L_3 + s_1c_2L_2 \\ s_{23}c_4 + c_{23}s_4 & -s_{23}s_4 + c_{23}c_4 & 0 & s_{23}L_3 + s_2L_2 + L_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

$$ \underbrace{ {}^{0}_{4}T = \begin{bmatrix} c_1c_{234} & -c_1s_{234} & s_1 & c_1(c_{23}L_3 + c_2L_2) \\ s_1c_{234} & -s_1s_{234} & -c_1 & s_1(c_{23}L_3 + c_2L_2) \\ s_{234} & c_{234} & 0 & L_1 + s_2L_2 + s_{23}L_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} } \Rightarrow \text{where} \begin{bmatrix} \text{Rotation Matrix } {}^{0}_{4}R & \text{Translation} \\ 0 \quad 0 \quad 0 & 1 \end{bmatrix} $$

This is the result. It allows us to calculate the rotation and translation of the end effector, given the joint angles.