

⑦

Python code for forward

Kinematics

```
1) import numpy as np.
    from numpy import array.
    from sympy import symbols,
        cos, sin, pi,
        simplify, sqrt,
        atan2, pprint
    from sympy.matrices import matrix.
```

# Creating symbols for DH parameter:

$q_1, q_2, q_3, q_4, q_5 = \text{symbols}('q1:6')$   
# (angles)

$d_1, d_2, d_3, d_4, d_5 = \text{symbols}('d1:8')$   
# link offsets.

$a_0, a_1, a_2, a_3, a_4 = \text{symbols}('a0:7')$  # link  
length.

$\alpha_0, \alpha_1, \alpha_2, \alpha_3, \alpha_4 = \text{symbols}('a0:7')$

$\alpha_0, \alpha_1, \alpha_2, \alpha_3, \alpha_4$

$= \text{symbols}('alpha0:7')$

# Joint twist angle.

# Creating DH table.

dh = {

	<u>alpha</u>	<u>a</u>	<u>d</u>	<u>q</u>
<del>alpha = 0</del>	0	0	0.75	$q_1$
$-\pi/2$	0.35	0		$q_2 - \pi/2$
0	1.25	0		$q_3$
$-\pi/2$	-0.034	1.5		$q_4$
$+\pi/2$	0	0		$q_5$
0 <del><math>\pi/2</math></del>	0	0		0.

}

# To get homogeneous Transformation matrix position & orient of E.E:

def TF\_Mat (alpha, a, d, q):

~~TF = matrix([ [ cos(q)~~

def ~~TF~~.

TF := matrix( [ [ cos(q) . . . , -sin(q) , 0, . . a ],

[ sin(q)\*cos(alpha), cos(q)\*cos(alpha) , -sin(alpha) ,  
- sin(alpha)\*d ],

[ sin(q)\*sin(alpha), cos(q)\*sin(alpha), cos(alpha) ,  
cos(alpha)\*d ],

[ 0 , 0 , 0 , 1 ] )

return TF .

# for DH Table (substitute).

T0 - 1 = TF - Mat ( alpha 0, a0, d1, q1). nils (du).

T1 - 2 = " " "

T2 - 3 = " "

T3 - 4 = " "

T5 - 6 = " "

# Transformation from Base to  
end effector: (F-k).



↑ link-2.

$$T_{0-2} = (T_{0-1} * T_{1-2}) \# (\text{Base}) \text{ link-0 to link-2}$$

$$T_{0-3} = (T_{0-2} * T_{2-3}) \# (\text{Base}) \text{ link-0 to link-3}$$

$$T_{0-4} =$$

$$T_{0-5} =$$

$$T_{0-6} = (T_{0-5} * T_{5-6}) \# (\text{Base}) \text{ link-0 to } \text{link-6} \\ \text{(end effector)}$$

orientation correction matrix.

~~R-y matrix.~~

$$R_y = \text{matrix} ([ [\cos(-np.pi/2), 0, \sin(-np.pi/2)]$$

$$[0, 1, 0, 0],$$

$$[-\sin(-np.pi/2), 0, \cos(-np.pi/2), 0]$$

$$[0, 0, 0, 1] ] )$$

$$R_z = \text{matrix} ([ [\cos(np.pi), -\sin(np.pi), 0, 0],$$