

ENME480 UR3e DH Table

1 Assignment of Axes

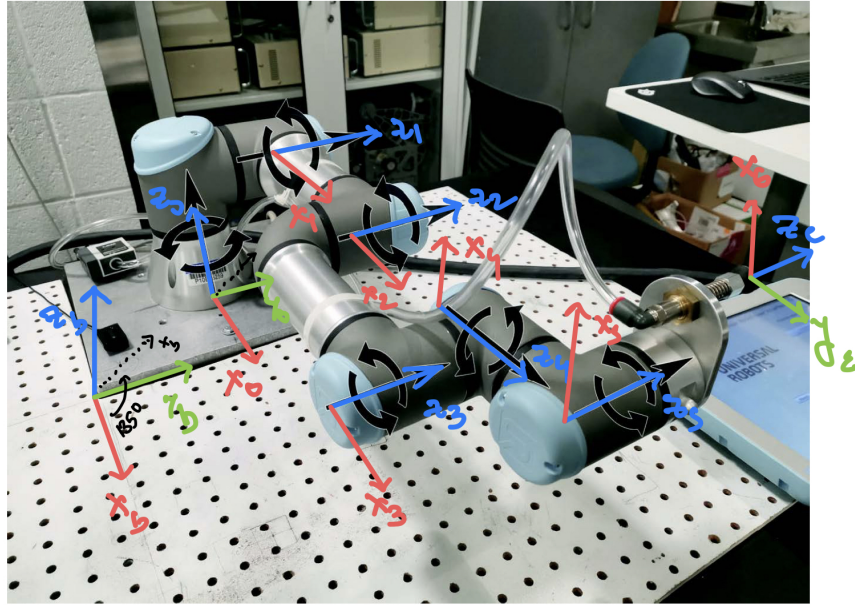


Figure 1: Axes are marked as per general convention. X-axis in red, and z-axis in blue

2 DH Table

Link	θ	α	a	d
b \rightarrow 0	135°	0	$0.15*\sqrt{2}$	0
0 \rightarrow 1	θ_1-135°	-90°	0	0.162
1 \rightarrow 2	θ_2	0	0.244	0.027
2 \rightarrow 3	θ_3	0	0.213	0
3 \rightarrow 4	θ_4-90°	-90°	0	0.083
4 \rightarrow 5	θ_5	90°	0	0.104
5 \rightarrow e	θ_6	0	0.0535	0.151

3 Python Function

```
1 def calculate_dh_transform(joint_positions):
2
3
4     ##### YOUR CODE STARTS HERE #####
5     # DH parameters for UR3e
6     # Modify these parameters according to the robot's configuration # a , alpha, d,
    theta
7     dh_params = [
8         [0.15*np.sqrt(2), 0.0, 0.0, 3*math.pi/4],
9         [0.0, -math.pi/2, 0.162, joint_positions[0]-3*math.pi/4], #
    theta1
10        [0.24365, 0.0, 0.0, joint_positions[1]], # theta2
11        [0.21325, 0.0, 0.0, joint_positions[2]], # theta3
12        [0.0, -math.pi / 2, 0.083, joint_positions[3]-math.pi/2], #
    theta4
13        [0.0, math.pi / 2, 0.104, joint_positions[4]], # theta5
14        [0.0535, 0.0, 0.151, joint_positions[5]] # theta6
15    ]
16
17    transform = np.eye(4)
18
19    for a, alpha, d, theta in dh_params:
20        transform_i = np.array([
21            [math.cos(theta), -math.sin(theta) * math.cos(alpha), math.sin(theta) * math
22            .sin(alpha), a * math.cos(theta)],
23            [math.sin(theta), math.cos(theta) * math.cos(alpha), -math.cos(theta) * math
24            .sin(alpha), a * math.sin(theta)],
25            [0, math.sin(alpha), math.cos(alpha), d],
26            [0, 0, 0, 1]
27        ])
28        transform = np.dot(transform, transform_i)
29
30    ##### YOUR CODE ENDS HERE #####
31    return transform
```

Listing 1: Python Function for calculating DH transform