ENME480 UR3e DH Table

1 Assignment of Axes

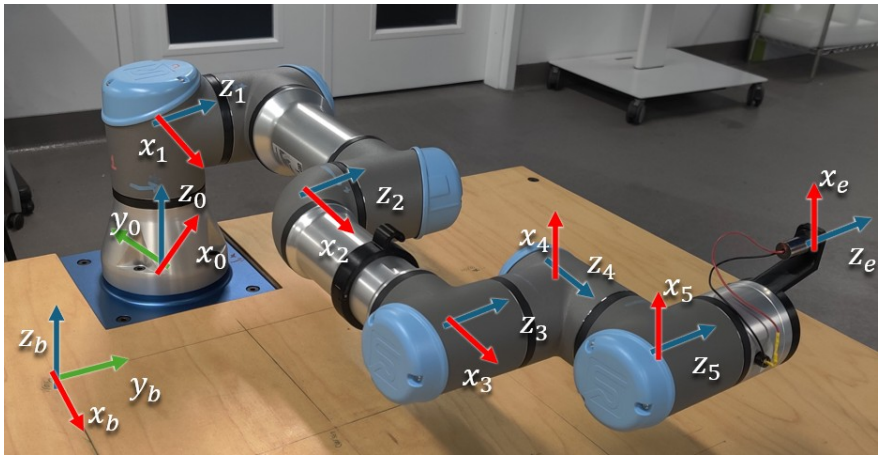

Figure 1: Axes are marked as per general convention. X-axis in red, and z-axis in blue

2 DH Table

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

3 Python Function

```python
def calculate_dh_transform(joint_positions):


        ################### YOUR CODE STARTS HERE #####################################
        # DH parameters for UR3e
        # Modify these parameters according to the robot's configuration # a , alpha, d, theta
        dh_params = [
            [0.15*np.sqrt(2),      0.0,     0.0,     3*math.pi/4],
            [0.0,               -math.pi/2,    0.162,     joint_positions[0]-3*math.pi/4],  # theta1
            [0.24365,           0.0,         0.0,     joint_positions[1]],  # theta2
            [0.21325,           0.0,         0.0,     joint_positions[2]],  # theta3
            [0.0,          -math.pi / 2,        0.083,       joint_positions[3]-math.pi/2],  # theta4
            [0.0,          math.pi / 2,       0.104,       joint_positions[4]],  # theta5
            [0.0535,         0.0,         0.151,     joint_positions[5]]   # theta6
        ]

        transform = np.eye(4)

        for a, alpha, d, theta in dh_params:
            transform_i = np.array([
                [math.cos(theta), -math.sin(theta) * math.cos(alpha), math.sin(theta) * math.sin(alpha), a * math.cos(theta)],
                [math.sin(theta), math.cos(theta) * math.cos(alpha), -math.cos(theta) * math.sin(alpha), a * math.sin(theta)],
                [0, math.sin(alpha), math.cos(alpha), d],
                [0, 0, 0, 1]
            ])
            transform = np.dot(transform, transform_i)


        ################################### YOUR CODE ENDS HERE ##########################

        return transform
```

Listing 1: Python Function for calculating DH transform