

Assignment 01: Forward Kinematics and Visualization of a Planar Robotic Arm

Name: Hardik

Roll Number: 240423

Task 1: Position Calculations

The base of the robotic arm is fixed at the origin.

- **Base:**(0,0)

Elbow Joint (x1,y1)

$$x1=l1\cos(q1)$$

$$y1=l1\sin(q1)$$

End-Effector (x,y)(x, y)(x,y)

$$x=l1\cos(q1)+l2\cos(q1+q2)$$

$$y=l1\sin(q1)+l2\sin(q1+q2)$$

Configuration 1: $q1=0^\circ$, $q2=0$ (STRAIGHT ARM)

$$\text{Elbow Joint } x1=1\cos(0^\circ) = 1$$

$$y1=1\sin(0^\circ) = 0$$

$$\text{End-Effector } x=1\cos(0^\circ)+1\cos(0^\circ) = 2$$

$$y=1\sin(0^\circ)+1\sin(0^\circ) = 0$$

Configuration 2: $q1=45^\circ$, $q2=45$ (BENT ELBOW)

$$\text{Elbow Joint } x1=1\cos(45^\circ)=0.707$$

$$y1=1\sin(45^\circ)=0.707$$

$$\text{End-Effector } x=1\cos(45^\circ)+1\cos(90^\circ)$$

$$x=0.707+0 = 0.707$$

$$y=1\sin(45^\circ)+1\sin(90^\circ)$$

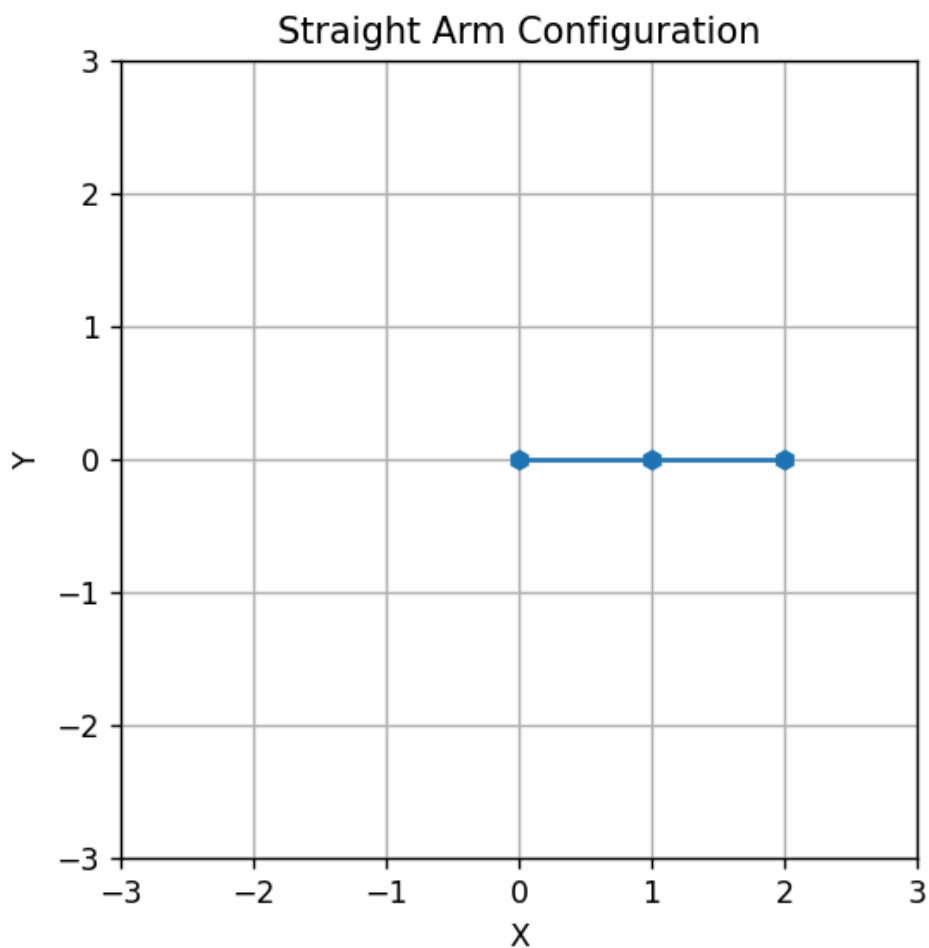
$$y=0.707+1=1.707$$

Here, $q1+q2=90$

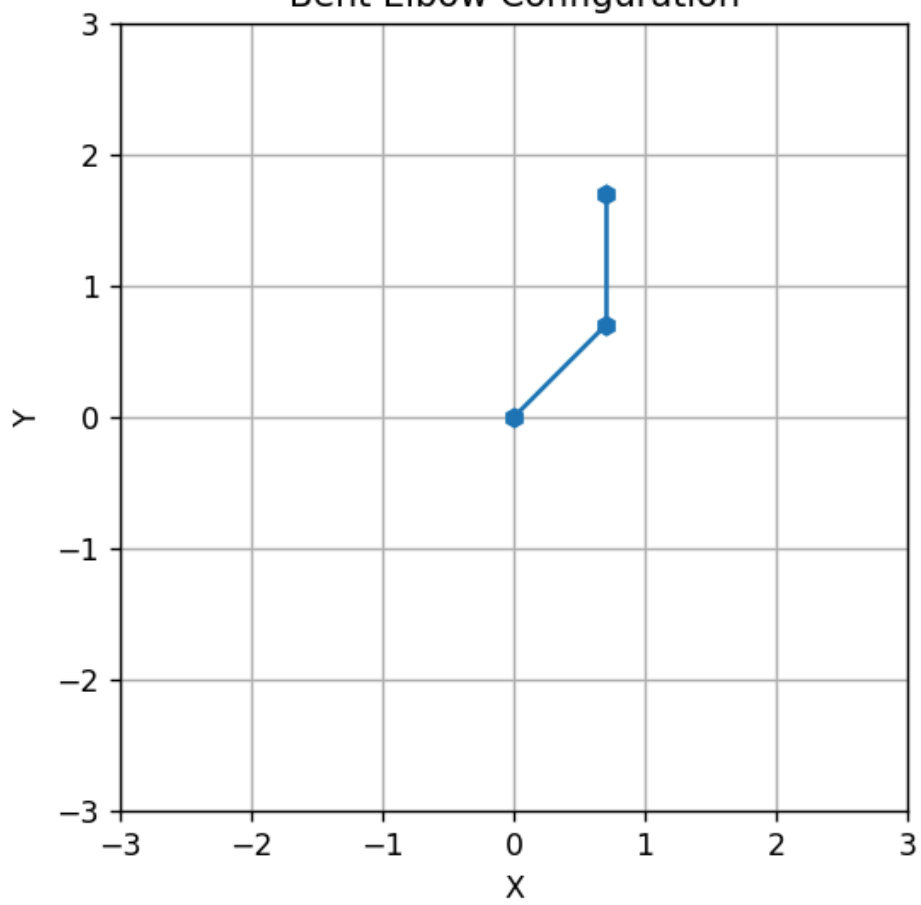
Configuration 3: $q_1=45^\circ$, $q_2=-90$ (FOLDED ARM)

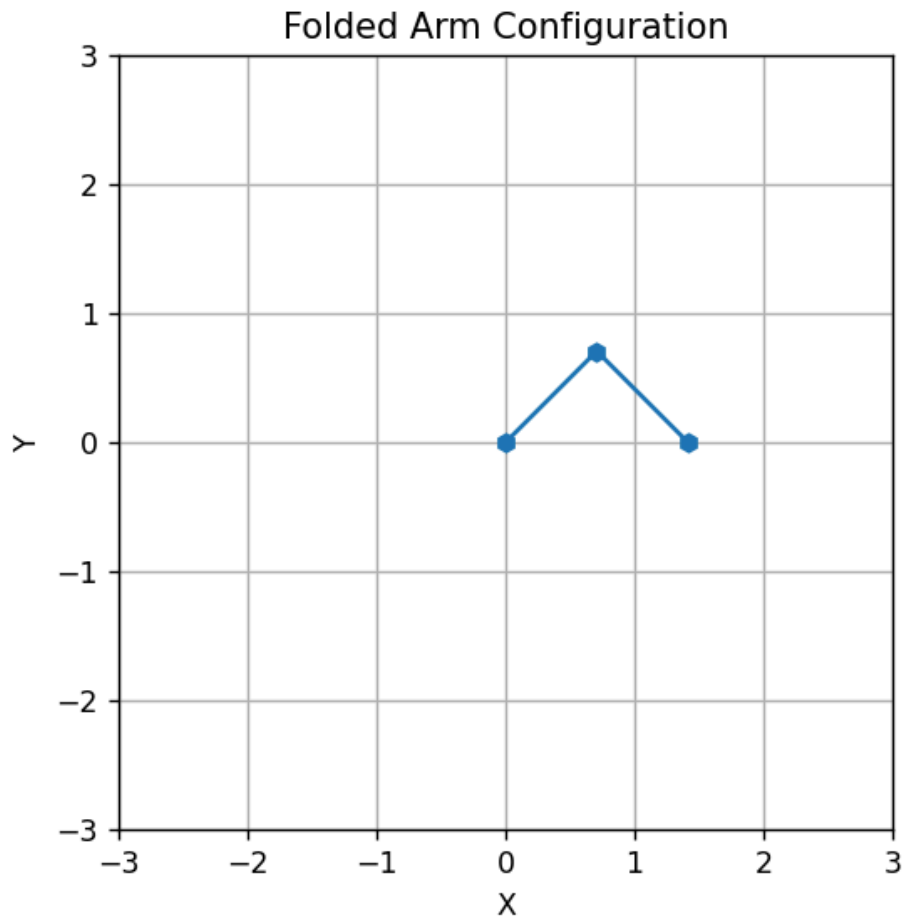
Elbow Joint	$x_1=1\cos(45^\circ)=0.707$	Here, $q_1+q_2=-45$
	$y_1=1\sin(45^\circ)=0.707$	
End-Effector	$x=1\cos(45^\circ)+1\cos(-45^\circ)$	
	$x=0.707+0.707=1.414$	
	$y=1\sin(45^\circ)+1\sin(-45^\circ)$	
	$y=0.707-0.707=0$	

TASK 2:



Bent Elbow Configuration





Effect of Joint Angles on Arm Position and Workspace

The joint angle q_1 controls the orientation of the entire robotic arm with respect to the base. Changing q_1 rotates both links together around the origin.

The joint angle q_2 controls the relative angle between the two links. Changing q_2 bends or folds the arm, directly affecting the distance and direction of the end-effector from the base.

By varying both q_1 and q_2 , the end-effector can reach a wide region in the plane. This reachable region is known as the workspace of the robotic arm.

PYTHON SCRIPT:

```
import numpy as np
import matplotlib.pyplot as plt

l1 = 1
l2 = 1

def forward_kinematics(q1, q2):
    x_e = l1 * np.cos(q1)
    y_e = l1 * np.sin(q1)
    x_ee = x_e + l2 * np.cos(q1 + q2)
    y_ee = y_e + l2 * np.sin(q1 + q2)
    return (x_e, y_e), (x_ee, y_ee)

def plot_arm(q1, q2, title):
    elbow, ee = forward_kinematics(q1, q2)

    x_points = [0, elbow[0], ee[0]]
    y_points = [0, elbow[1], ee[1]]

    plt.figure()
    plt.plot(x_points, y_points, '-h')
    plt.xlim(-3, 3)
    plt.ylim(-3, 3)
    plt.xlabel("X")
    plt.ylabel("Y")
    plt.title(title)
    plt.grid(True)
    plt.gca().set_aspect('equal')
    plt.show()

q1 = 0
q2 = 0
plot_arm(q1, q2, "Straight Arm Configuration")

q1 = np.pi / 4
q2 = np.pi / 4
plot_arm(q1, q2, "Bent Elbow Configuration")

q1 = np.pi / 4
q2 = -np.pi / 2
plot_arm(q1, q2, "Folded Arm Configuration")
```

