Assignment -1 : Two-Link Planar RoboticArm Analysis

This assignment studies a two-link planar robotic arm with two revolute joints. The arm is fixed at the origin and operates in a two-dimensional Cartesian plane. Each link has equal length ($l1 = l2 = 1$). The goal is to compute joint positions using forward kinematics and visualize the arm for different configurations.

Forward Kinematics

The configuration of the robotic arm is defined by two joint angles q1andq2.

Elbow joint position:

$x = l1 \cos(q1)$

$y = l1 \sin(q1)$

End-effector position:

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

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

Configurations Analyzed

Three different arm configurations were studied:

1. **Straight Arm**

   $q1 = 0$, $q2 = 0$
2. **Bent Elbow**

   $q1 = \pi/4$, $q2 = \pi/4$

3. **Folded Arm**

$q1 = \pi/2, q2 = -\pi/2$

These configurations show how joint angles affect arm posture.

Visualization

The robotic arm was visualized using a 2D plot: · Base at origin

- **First link from base to elbow**
- **Second link from elbow to end effector**

Equal axis scaling was used to maintain correct geometry.

Workspace Analysis

The workspace is the set of all points reachable by the end effector. By varyingq1 and q2 between −π and π, the arm can reach all points inside a circular region with radius 2 units. The workspace was visualized using scatter plots.

Conclusion

This assignment demonstrates the relationship between joint angles and end effector position in a planar robotic arm. Forward kinematics equations were implemented successfully, and different configurations and workspace visualization provided insight into robotic motion and reachability.

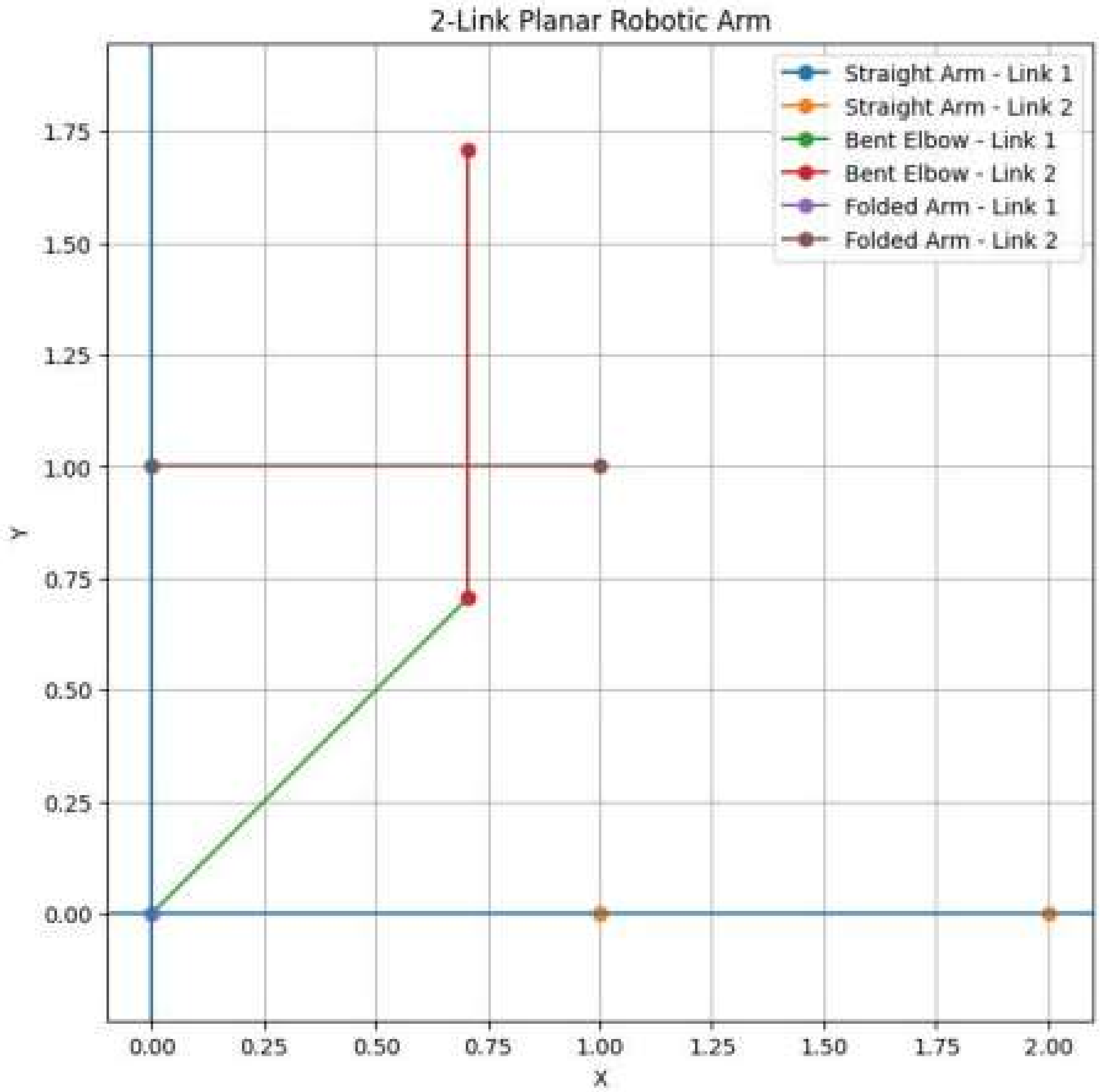
Code :

```
import numpy as np

import matplotlib.pyplot as plt
```

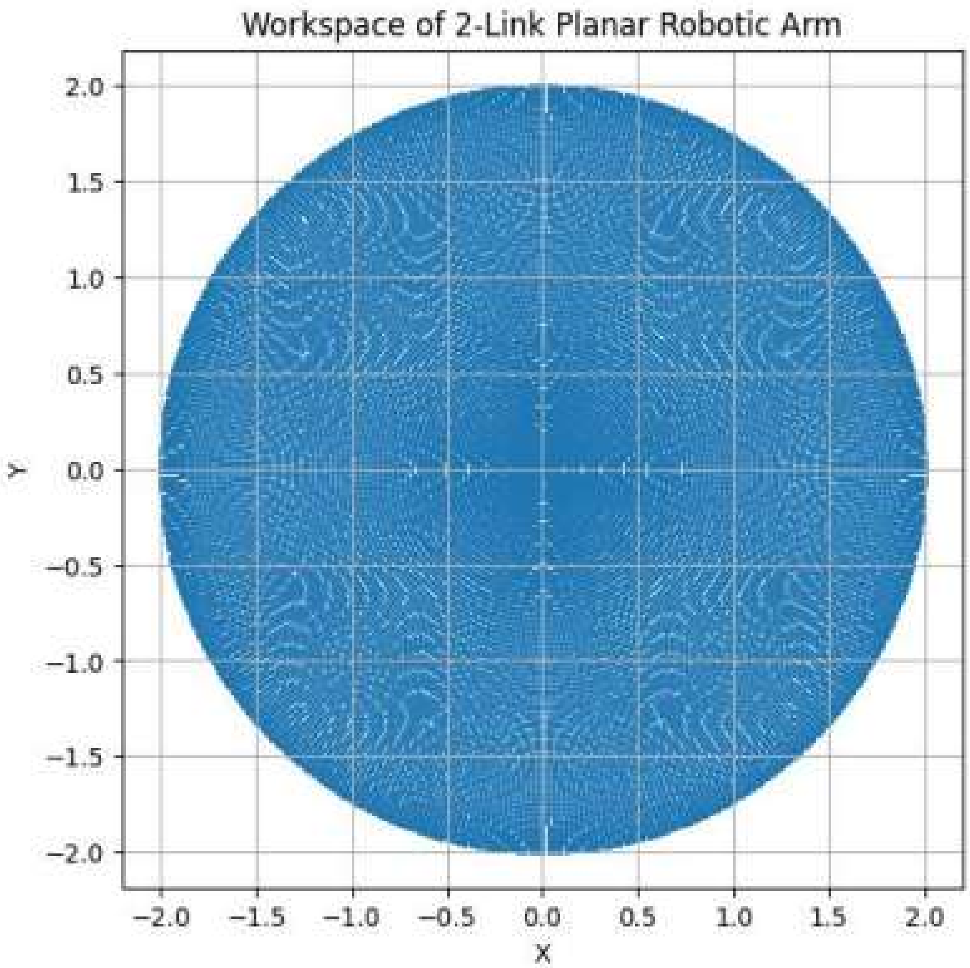
```
# Link lengths

l1 = 1

l2 = 1


def forward_kinematics(q1, q2):

    # Elbow position

    x_elbow = l1 * np.cos(q1)

    y_elbow = l1 * np.sin(q1)


    # End effector position x_ee = x_elbow + l2 * np.cos(q1 + q2)

    y_ee = y_elbow + l2 * np.sin(q1 + q2)


    return x_elbow, y_elbow, x_ee, y_ee


# Configurations

configs = {

    "Straight Arm": (0, 0),

    "Bent Elbow": (np.pi/4, np.pi/4),

    "Folded Arm": (np.pi/2, -np.pi/2)
}


plt.figure(figsize=(8, 8))
```

```
for name, (q1, q2) in configs.items():
    x_elbow, y_elbow, x_ee, y_ee = forward_kinematics(q1, q2)

    # Plot links
    plt.plot([0, x_elbow], [0, y_elbow], 'o-', label=f'{name} - Link 1') plt.plot([x_elbow, x_ee],
    [y_elbow, y_ee], 'o-', label=f'{name} - Link 2')

    # Print positions
    print(f"{name}")
    print(f" Elbow Position: ({x_elbow:.2f}, {y_elbow:.2f})")
    print(f" End Effector Position: ({x_ee:.2f}, {y_ee:.2f})\n")

plt.axhline(0)
plt.axvline(0)
plt.xlabel("X")
plt.ylabel("Y")
plt.title("2-Link Planar Robotic Arm")
plt.legend()
plt.grid()
plt.axis("equal")
plt.show()
```

```
# Workspace Visualization

q1_vals = np.linspace(-np.pi, np.pi, 200) q2_vals =
np.linspace(-np.pi, np.pi, 200)

x_ws = []
y_ws = []

for q1 in q1_vals:
    for q2 in q2_vals:
        x = l1*np.cos(q1) + l2*np.cos(q1 + q2) y = l1*np.sin(q1) +
        l2*np.sin(q1 + q2) x_ws.append(x)
        y_ws.append(y)

plt.figure(figsize=(6,6))
plt.scatter(x_ws, y_ws, s=1)
plt.xlabel("X")
plt.ylabel("Y")
plt.title("Workspace of 2-Link Planar Robotic Arm") plt.axis("equal")
plt.grid()
plt.show()
```



2-Link Planar Robotic Arm
Straight Arm - Link 1
Straight Arm - Link 2
Bent Elbow - Link 1
Bent Elbow - Link 2
Folded Arm - Link 1
Folded Arm - Link 2
Y
X
0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75
2.00



Jaypal (240494)