The robotic arm consists of two links of length $l1 = 1$and $l2 = 1$. The configuration is defined by joint angles q1 and q2.

Task 1: Position Calculations

To visualize the arm, we calculate the coordinates as follows:

- **Base:** Fixed at the origin (0, 0).
- Elbow Joint (x1, y1):

  $x1 = l1 \cos(q1)$

  $y1 = l1 \sin(q1)$

- End-Effector (x, y):

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

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

Straight Arm (q1=0°, q2=0°):

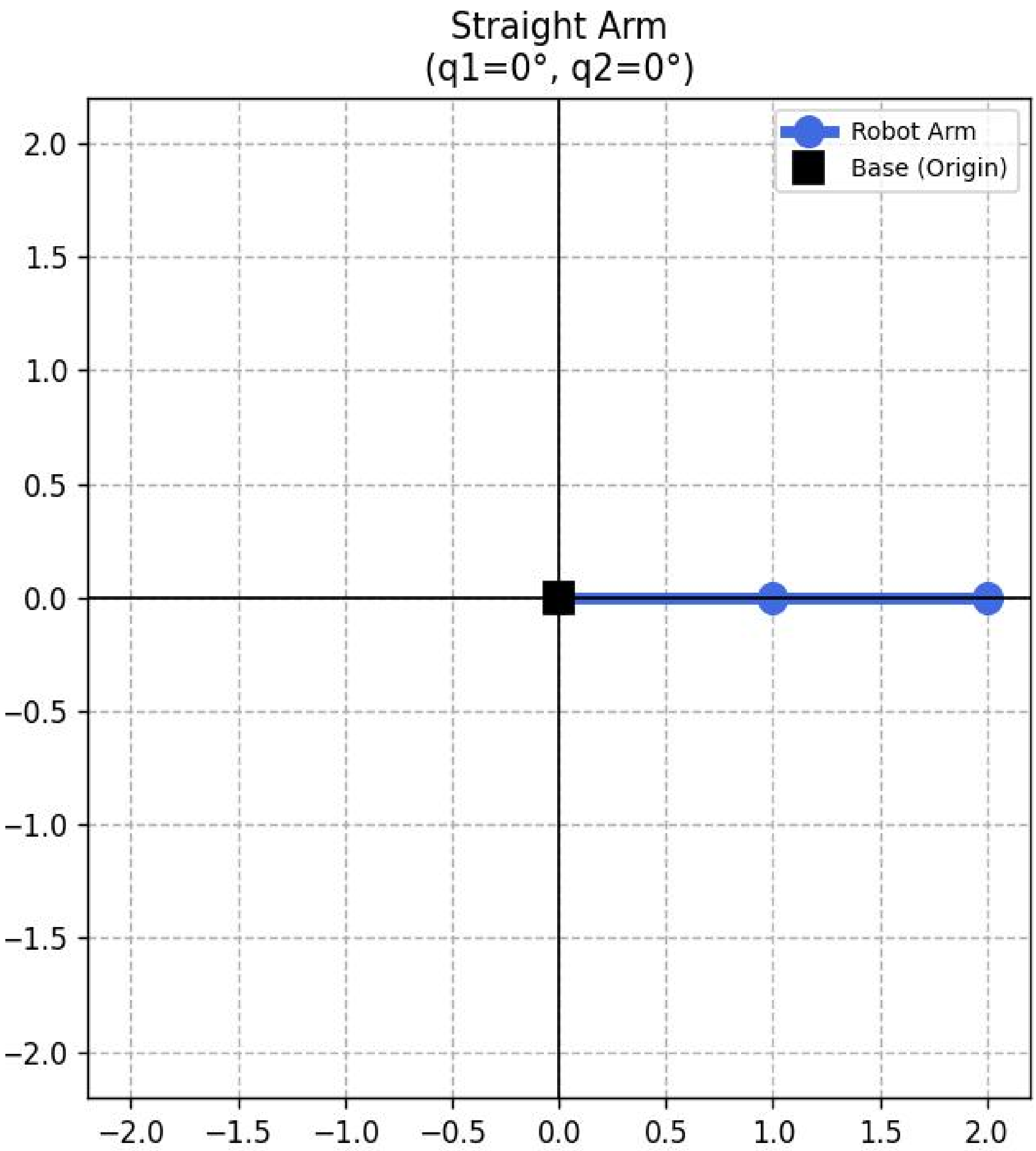
- Elbow: x1 =1cos(0) = 1, y1 = 1sin(0^) = 0
- End-Effector: x = 1cos(0) + 1cos(0) = 2, y = 1sin(0) + 1sin(0) = 0

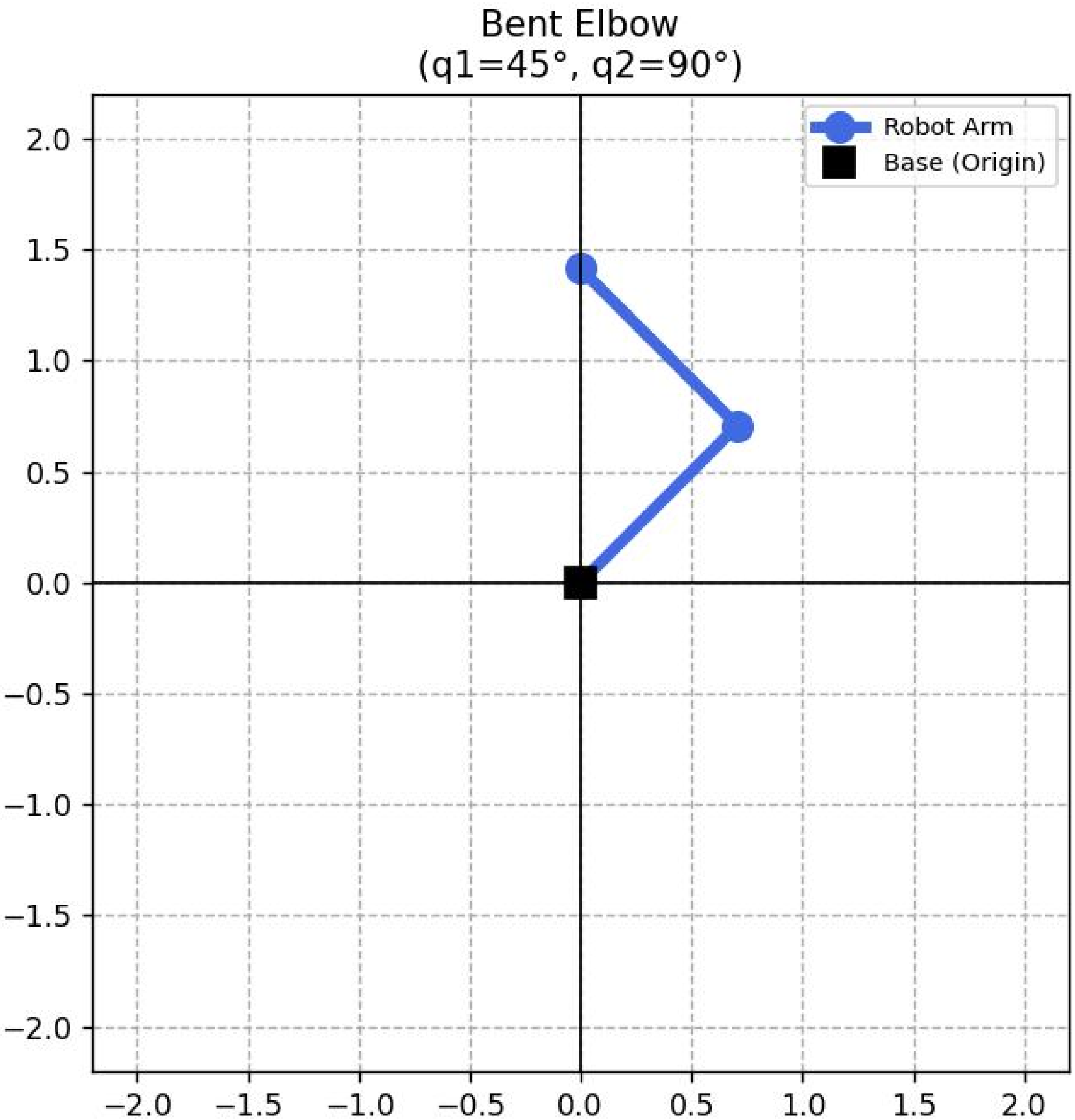
Bent Elbow (q1=45°, q2=90°):

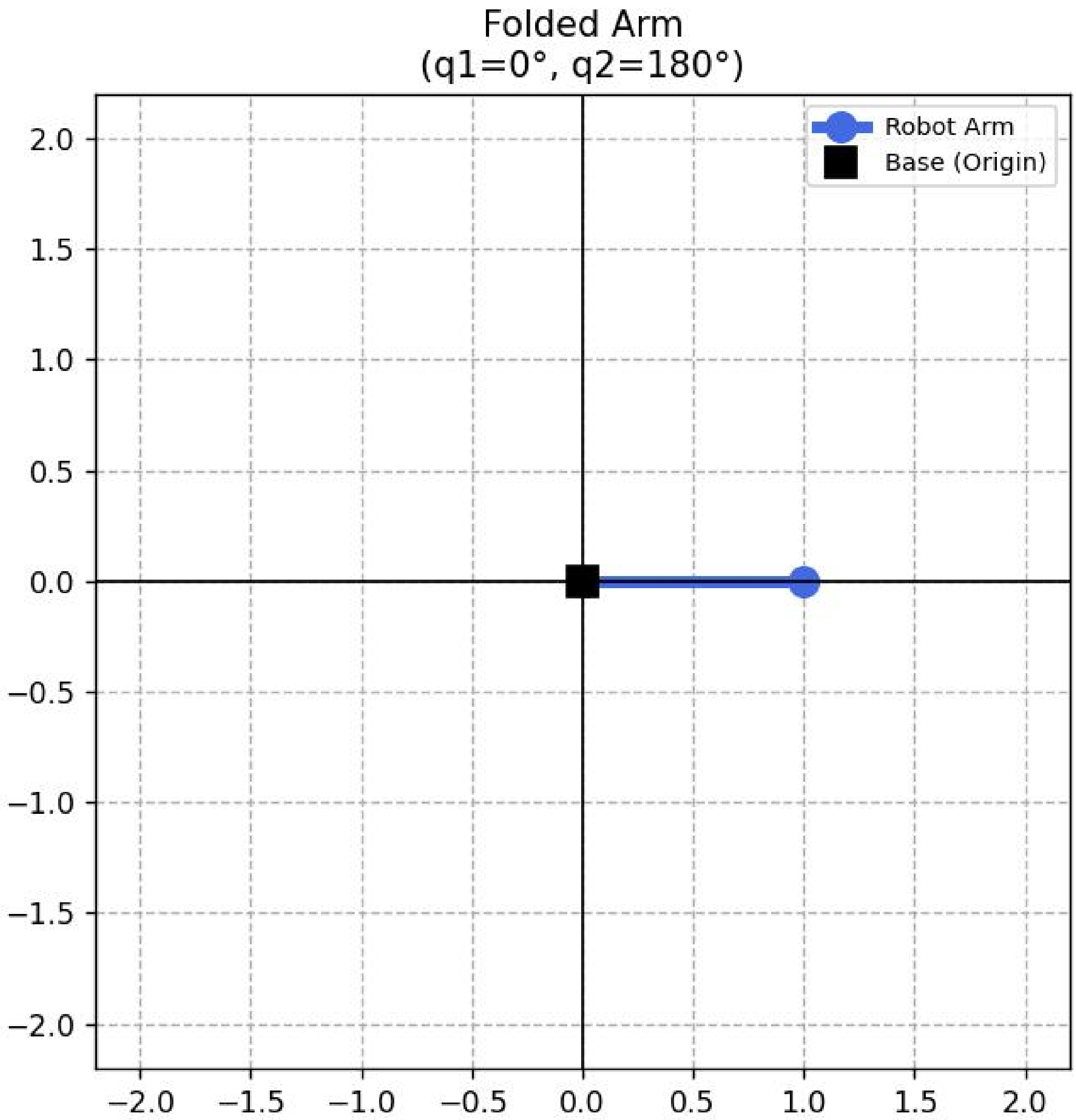
- Elbow: x1 = 0.707, y1 = 0.707
- End-Effector: x = cos(45°) + cos(135°) = 0, y = sin(45°) + sin(135°) = 1.414

Folded Arm (q1=0°, q2=180°):

- Elbow: x1 = 1, y1 = 0
- End-Effector: x = cos(0) + cos(180°) = 0, y = sin(0) + sin(180°) = 0


Straight Arm
(q1=0°, q2=0°)
Robot Arm
Base (Origin)
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−2.0
−2.0
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0


Bent Elbow
(q1=45°, q2=90°)
Robot Arm
Base (Origin)
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−2.0
−2.0
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0


Folded Arm
(q1=0°, q2=180°)
Robot Arm
Base (Origin)
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−2.0
−2.0
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0

Task 3:

Python script-

```
import numpy as np

import matplotlib.pyplot as plt

def plot_robotic_arm(q1_deg, q2_deg, ax, title):

   l1 = 1

   l2 = 1

   q1 = np.radians(q1_deg)

   q2 = np.radians(q2_deg)

   x1 = l1 * np.cos(q1)

   y1 = l1 * np.sin(q1)

   x2 = l1 * np.cos(q1) + l2 * np.cos(q1 + q2)

   y2 = l1 * np.sin(q1) + l2 * np.sin(q1 + q2)

   ax.plot([0, x1, x2], [0, y1, y2], marker='o', markersize=10, linewidth=4, color='royalblue',
label='Robot Arm')

   ax.plot(0, 0, 'ks', markersize=10, label='Base (Origin)')

   ax.set_title(f"{title}\n(q1={q1_deg}°, q2={q2_deg}°)", fontsize=12)

   ax.set_xlim(-2.2, 2.2)

   ax.set_ylim(-2.2, 2.2)

   ax.set_aspect('equal')

   ax.grid(True, linestyle='--')

   ax.axhline(0, color='black', lw=1)

   ax.axvline(0, color='black', lw=1)

   ax.legend(loc='upper right', fontsize='small')

fig, axs = plt.subplots(1, 3, figsize=(18, 6))

plot_robotic_arm(0, 0, axs[0], "Straight Arm")

plot_robotic_arm(45, 90, axs[1], "Bent Elbow")

plot_robotic_arm(0, 180, axs[2], "Folded Arm")

plt.tight_layout()

plt.show()

configs = [(0, 0), (45, 90), (0, 180)]
```

```
print("Coordinate Results for Task 1:")
for q1, q2 in configs:
  r1, r2 = np.radians(q1), np.radians(q2)
  ex, ey = np.cos(r1), np.sin(r1)
  eex = ex + np.cos(r1 + r2)
  eey = ey + np.sin(r1 + r2)
  print(f"Angles({q1},{q2}) -> Elbow: ({ex:.2f}, {ey:.2f}), End-Effector: ({eex:.2f}, {eey:.2f})")
```

Analysis of Arm Position and Workspace

The configuration of the 2-link planar robotic arm is governed by the two joint angles, q1 and q2, which determine the position of the elbow and the end-effector in the Cartesian plane.

1. Effect of Joint Angles on Position

- **Angle q1 (Base Joint):** This angle controls the rotation of the entire arm assembly relative to the origin. Changing q1 moves the arm in a circular arc around the base, determining the direction in which the arm points.
- **Angle q2 (Elbow Joint):** This angle defines the orientation of the second link relative to the first. It primarily affects the "extension" or "reach" of the arm.
  - When q2 = 0°, the arm is **fully extended** (straight), reaching its maximum distance from the origin.
  - When q2 = 180°, the arm is **folded back**, bringing the end-effector back toward the base.

2. Workspace Analysis

- **Reachable Workspace:** Since the links are of equal length (l1 = l2 = 1), the total maximum reach is l1 + l2 = 2.
- **Geometry:** The workspace is a **solid disk** centered at the origin with a radius of 2.
- **Accessibility:** Because the links are equal, the arm can reach any point from the boundary (r = 2) all the way down to the origin (r = 0), providing a full range of motion within that circular area.