DEPARTMENT OF MATHEMATICS

Mathematics Lab Assessment No. 4

<table>
<tr><td colspan="2">Name: Srushti g joshi
SRN: 01FE24BAR014
Date:4/12/2025</td></tr>
<tr><td></td><td>Problem on Robot 2R:
Planar Robot 2R with two link lengths $L_1 = 5\ m$, $L_2 = 3\ m$ and joint angles $\theta_1 = 45°, \theta_2 = 30°$. Use homogeneous transformation matrix to compute the end effector position. Plot and study the work space for the joint limits $\theta_{1min} = 0°, \theta_{1max} = 180°, \theta_{2min} = -90°, \theta_{2max} = 90°$ and 120 sampling points.

Problem on Robot 3R:
Planar Robot 3R with two link lengths $L_1 = 5\ m$, $L_2 = 3\ m, L_3 = 2\ m$ and joint angles $\theta_1 = 45°, \theta_2 = 30°, \theta_3 = -30°$. Use homogeneous transformation matrix to compute the end effector position. Plot and study the work space for the joint limits $\theta_{1min} = 0°, \theta_{1max} = 180°, \theta_{2min} = -90°, \theta_{2max} = 90°, \theta_{3min} = -90°, \theta_{3max} = 90°$ and 30 sampling points.</td></tr>
<tr><td>Solution</td><td>i) Identify the parameters and mathematical concept

Robot Type : Planar 2R & 3R Serial Manipulator
Unknown : End effector position (x, y)
Method : Forward Kinematics + Homogeneous Transformation
Output : Workspace Plot</td></tr>
<tr><td></td><td>ii) Solve analytically

2R Robot
End-effector position:
x = l1 cosθ1 + l2 cos(θ1 + θ2)
y = l1 sinθ1 + l2 sin(θ1 + θ2)

3R Robot
x = l1 cosθ1 + l2 cos(θ1 + θ2) + l3 cos(θ1 + θ2 + θ3)
y = l1 sinθ1 + l2 sin(θ1 + θ2) + l3 sin(θ1 + θ2 + θ3)

End-Effector Position at Given Angles:
x = l1cos45 + l2cos75
y = l1sin45 + l2sin75

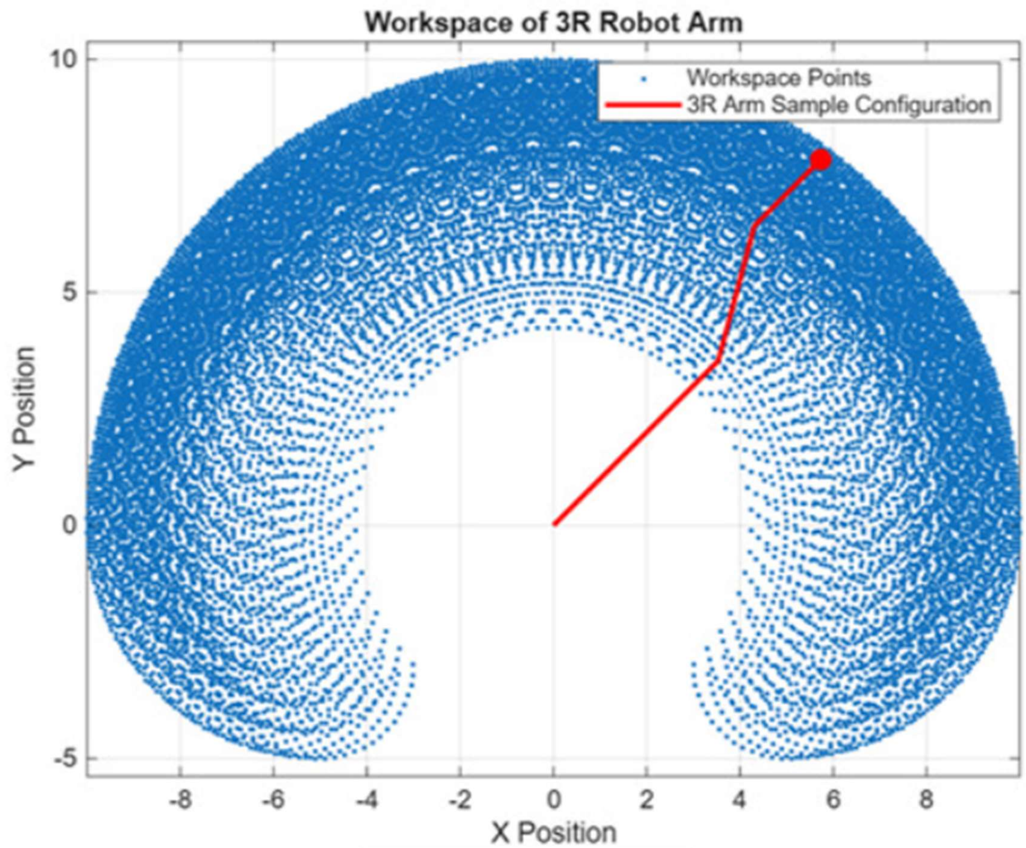
2R Robot at θ1 = 45°, θ2 = 30°
x ≈ 5.94 m
y ≈ 5.75 m</td></tr>
</table>

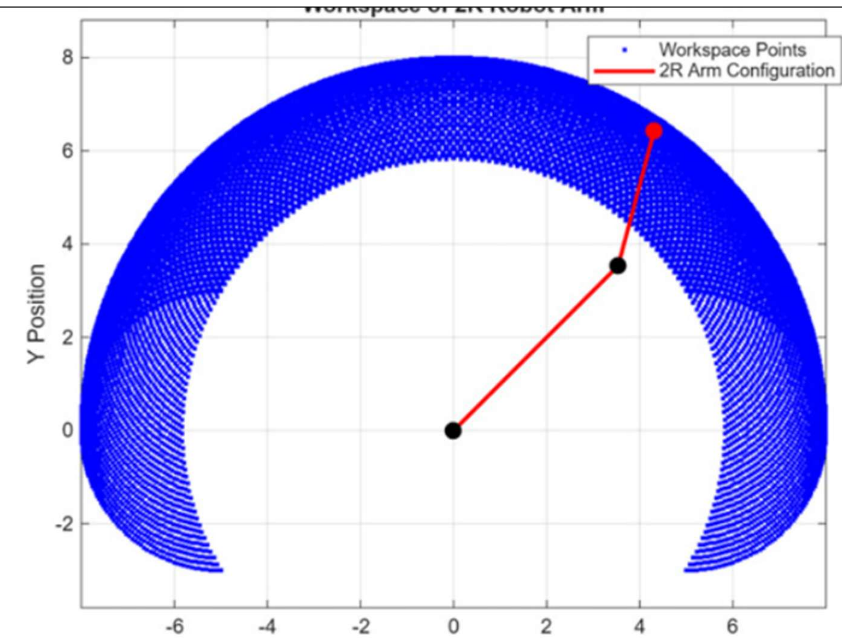
3R Robot at θ1 = 45°, θ2 = 30°, θ3 = 80°
x ≈ 4.74 m
y ≈ 5.51 m

iii) GeoGebra Screenshot / Program Execution

CODE:

```
import numpy as np
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
import scipy.linalg as la

# Link lengths
L1, L2, L3 = 50, 50, 50

# Joint angle limits (radians)
theta1_min, theta1_max = 0, np.pi
theta2_min, theta2_max = -np.pi/2, np.pi/2
theta3_min, theta3_max = -np.pi/2, np.pi/2

# Sampling resolution
num_points = 30

# Generate joint angle vectors for workspace sampling
theta1 = np.linspace(theta1_min, theta1_max, num_points)
theta2 = np.linspace(theta2_min, theta2_max, num_points)
theta3 = np.linspace(theta3_min, theta3_max, num_points)

# Compute workspace points by forward kinematics
workspace_x, workspace_y = [], []
for t1 in theta1:
    for t2 in theta2:
        for t3 in theta3:
            x = L1*np.cos(t1) + L2*np.cos(t1 + t2) + L3*np.cos(t1 + t2 + t3)
            y = L1*np.sin(t1) + L2*np.sin(t1 + t2) + L3*np.sin(t1 + t2 + t3)
            workspace_x.append(x)
            workspace_y.append(y)

# Plot workspace
plt.figure()
plt.plot(workspace_x, workspace_y, '.', markersize=5)
plt.axis('equal')
plt.grid(True)
plt.title('Workspace of 3R Robot Arm')
plt.xlabel('X Position')
plt.ylabel('Y Position')

# Plot sample robot arm configuration
sample_theta = [np.pi/2, np.pi/2, -np.pi/2]
joint1 = np.array([0, 0])
joint2 = joint1 + np.array([L1*np.cos(sample_theta[0]), L1*np.sin(sample_theta[0])])
joint3 = joint2 + np.array([L2*np.cos(sample_theta[0]+sample_theta[1]), L2*np.sin(sample_theta[0]+sample_theta[1])])
end_effector = joint3 + np.array([L3*np.cos(sum(sample_theta)), L3*np.sin(sum(sample_theta))])
```

	iv) Results and analysis from the graph  **OUTPUT :**



Maximum reach = l1 + l2 = 8 m
Minimum reach = |l1 − l2| = 2 m