Table of Contents

QUESTION 5:

```
% Here we are changing p2 to 7 and resolving problem 4

% Plotting f(theta) on [-pi, pi]
theta_vals = -pi:0.01:pi;

f_vals = f_variable_p2(theta_vals, 7); % p2=7

figure(10)
plot(theta_vals, f_vals)
xlabel('\theta (radians)')
ylabel('f(\theta)')
title('Plot of f(\theta) on [-\pi, \pi] for Question #5')
yline(0, '--r');
drawnow;

% The problem states that there are now six poses

% Finding the six theta values (guesses are from eyeballing the graph)
p2 = 7;
f_p2 = @(theta) f_variable_p2(theta, p2);

theta1 = fzero(f_p2, -0.68);
theta2 = fzero(f_p2, -0.36);
theta3 = fzero(f_p2, 0.03);
theta4 = fzero(f_p2, 0.44);
theta5 = fzero(f_p2, 0.97);
theta6 = fzero(f_p2, 2.5);

thetas = [theta1 theta2 theta3 theta4 theta5 theta6];

% From the above it appears that our roots are at:
% theta = -0.6732, -0.3547, 0.0378, 0.4589, 0.9777, and 2.5139 rad
figure(11)
plot(theta_vals, f_vals)
xlabel('\theta (radians)')
ylabel('f(\theta)')
title('Plot of f(\theta) on [-\pi, \pi] with roots')
yline(0, '--r');
xline(theta1, '--r', '-0.6732');
xline(theta2, '--r', '-0.3547');
xline(theta3, '--r', '0.0378');
xline(theta4, '--r', '0.4589');
xline(theta5, '--r', '0.9777');
xline(theta6, '--r', '2.5139');
```

```
drawnow;

% Since we're asked to solve the forward kinematics problem, we need to
% solve for x and y now (we just solved for theta)

% Finding the x and y coordinates for the four poses
[x_1 y_1] = forward_kinematics_variable_p2(theta1, p2);
[x_2 y_2] = forward_kinematics_variable_p2(theta2, p2);
[x_3 y_3] = forward_kinematics_variable_p2(theta3, p2);
[x_4 y_4] = forward_kinematics_variable_p2(theta4, p2);
[x_5 y_5] = forward_kinematics_variable_p2(theta5, p2);
[x_6 y_6] = forward_kinematics_variable_p2(theta6, p2);

xs = [x_1 x_2 x_3 x_4 x_5 x_6];
ys = [y_1 y_2 y_3 y_4 y_5 y_6];

% It was found that
% (x_1, y_1) = (-4.3148, 2.5264)
% (x_2, y_2) = (-4.8049, 1.3831)
% (x_3, y_3) = (-4.9490, 0.7121)
% (x_4, y_4) = (-0.8198, 4.9323)
% (x_5, y_6) = (2.3036, 4.4378)
% (x_5, y_6) = (3.2157, 3.8287)

% Now we need to plot the four poses
% Helper function is in the supporting functions section

for i = 1:6
    draw_pose(11+i, xs(i), ys(i), thetas(i), 5, i);
    drawnow;
end

% The strut lengths are correct!!!
```

ALL FUNCTIONS SUPPORTING THIS CODE

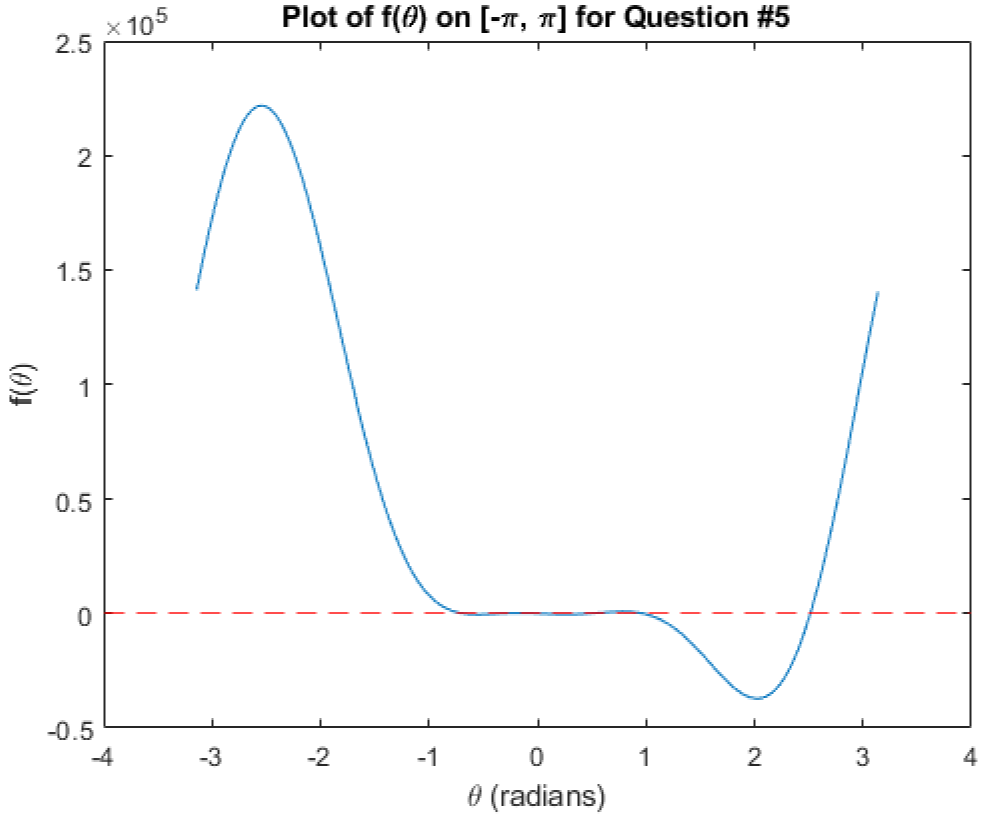
f(theta) function with ability to change p2

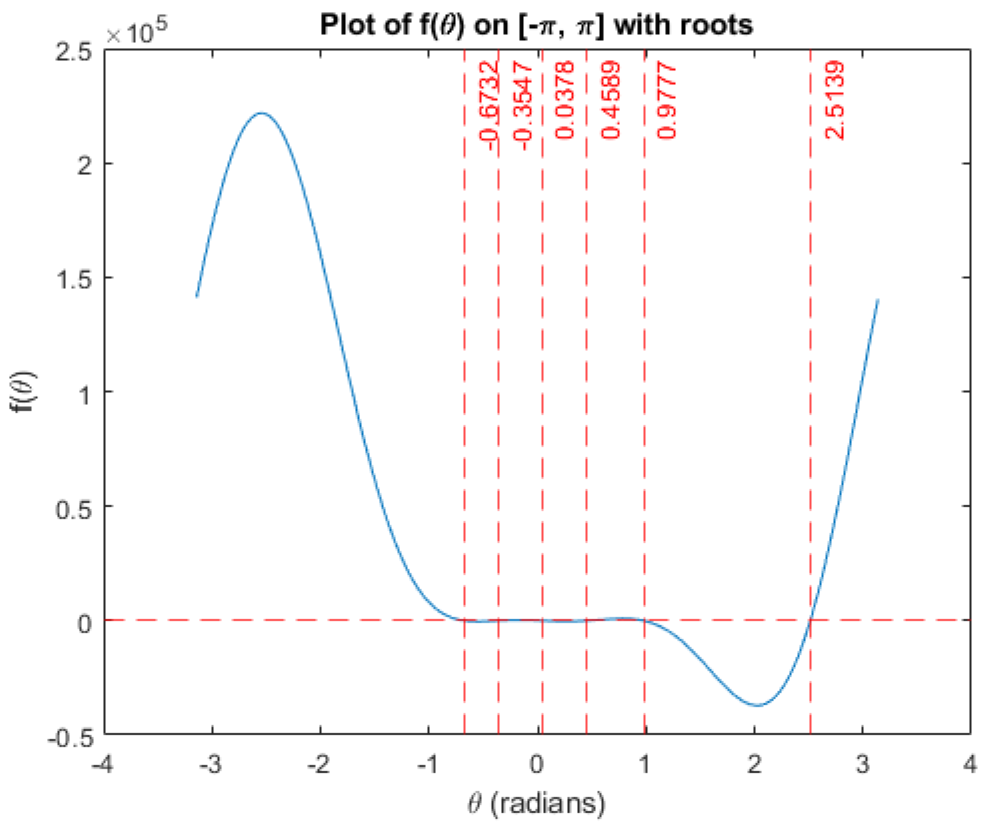
```
function out = f_variable_p2(theta, p2)
    L1 = 3; L2 = 3 * sqrt(2); L3 = 3;
    gamma = pi / 4;
    p1 = 5; p3 = 3;
    x1 = 5; x2 = 0; y2 = 6;

    A2 = L3 * cos(theta) - x1;
    B2 = L3 * sin(theta);
    A3 = L2 * (cos(theta) * cos(gamma) - sin(theta) * sin(gamma)) - x2;
    B3 = L2 * (cos(theta) * sin(gamma) + sin(theta) * cos(gamma)) - y2;

    N1 = B3 .* (p2^2 - p1^2 - A2.^2 - B2.^2) - B2 .* (p3^2 - p1^2 - A3.^2 -
B3.^2);
    N2 = -A3 .* (p2^2 - p1^2 - A2.^2 - B2.^2) + A2 .* (p3^2 - p1^2 - A3.^2 -
B3.^2);
```

```
    D = 2 * (A2 .* B3 - B2 .* A3);

    out = N1.^2 + N2.^2 - p1.^2 * D.^2;
end

% Forward kinematics problem solver with variable p2
function [x, y] = forward_kinematics_variable_p2(theta, p2)

    % Platform lengths
    L1 = 3;
    L2 = 3 * sqrt(2);
    L3 = 3;

    % Angle across from L1
    gamma = pi / 4;

    % Strut lengths
    p1 = 5;
    p3 = 3;

    % Strut base positions
    x1 = 5;
    x2 = 0;
    y2 = 6;

    % Compute intermediate terms
    A2 = L3 * cos(theta) - x1;
    B2 = L3 * sin(theta);
    A3 = L2 * (cos(theta) * cos(gamma) - sin(theta) * sin(gamma)) - x2;
    B3 = L2 * (cos(theta) * sin(gamma) + sin(theta) * cos(gamma)) - y2;

    % Numerators and denominator
    N1 = B3 .* (p2^2 - p1^2 - A2.^2 - B2.^2) - B2 .* (p3^2 - p1^2 - A3.^2 -
B3.^2);
    N2 = -A3 .* (p2^2 - p1^2 - A2.^2 - B2.^2) + A2 .* (p3^2 - p1^2 - A3.^2 -
B3.^2);
    D = 2 * (A2 .* B3 - B2 .* A3);

    % Solve for x and y
    x = N1 / D;
    y = N2 / D;

end


function draw_pose(fig_num, x, y, theta, question_number, pose_index)
    % Constants
    L2 = 3 * sqrt(2);
    L3 = 3;
    gamma = pi/4;
    x1 = 5; x2 = 0; y2 = 6;

    % Triangle corner positions
    u1 = x;
```

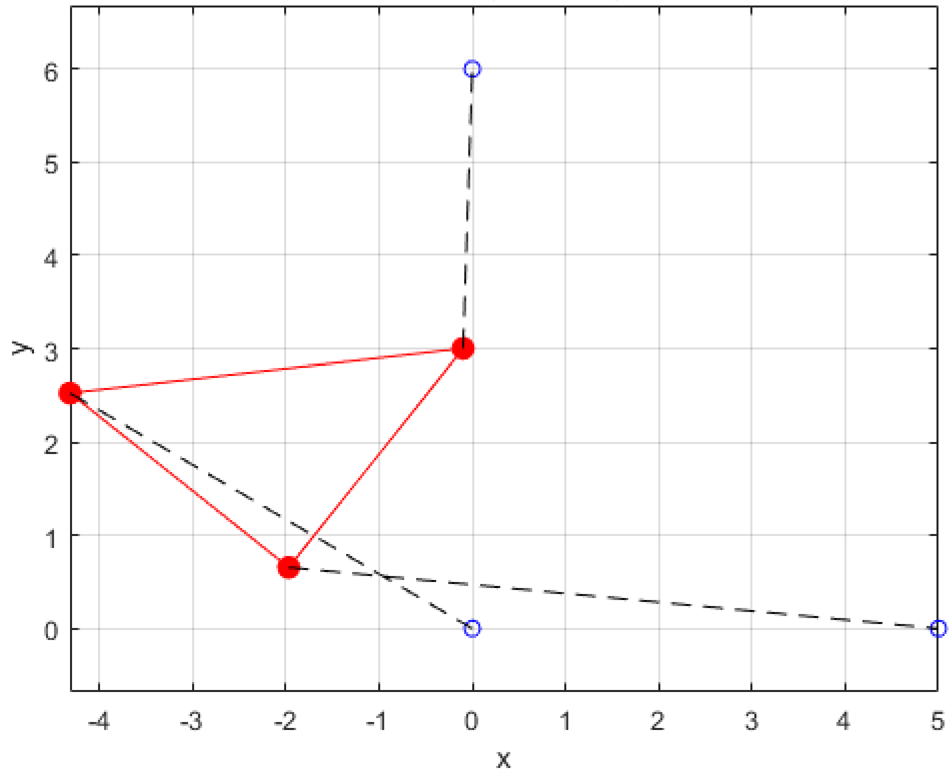
```
    v1 = y;
    u2 = x + L3 * cos(theta);
    v2 = y + L3 * sin(theta);
    u3 = x + L2 * cos(theta + gamma);
    v3 = y + L2 * sin(theta + gamma);

    % Compute strut lengths
    p1 = norm([u1, v1] - [0, 0]);
    p2 = norm([u2, v2] - [x1, 0]);
    p3 = norm([u3, v3] - [x2, y2]);

    % Plot
    figure(fig_num)
    plot([u1 u2 u3 u1], [v1 v2 v3 v1], 'r'); hold on
    plot([0 x1 x2], [0 0 y2], 'bo')
    plot([u1 u2 u3], [v1 v2 v3], 'ro', 'MarkerSize', 8, 'MarkerFaceColor',
'r')
    plot([u1 0], [v1 0], 'k--')    % p1
    plot([u2 x1], [v2 0], 'k--')   % p2
    plot([u3 x2], [v3 y2], 'k--')  % p3

    % Pose label
    title_str = sprintf('Pose %d (\\theta = %.2f)', pose_index, theta);


    title(title_str)
    xlabel('x')
    ylabel('y')
    axis equal
    grid on

    % Print strut lengths
    fprintf("Pose %d: p1 = %.4f, p2 = %.4f, p3 = %.4f\n", pose_index, p1,
p2, p3);
end
```

```
Pose 1: p1 = 5.0000, p2 = 7.0000, p3 = 3.0000
Pose 2: p1 = 5.0000, p2 = 7.0000, p3 = 3.0000
Pose 3: p1 = 5.0000, p2 = 7.0000, p3 = 3.0000
Pose 4: p1 = 5.0000, p2 = 7.0000, p3 = 3.0000
Pose 5: p1 = 5.0000, p2 = 7.0000, p3 = 3.0000
Pose 6: p1 = 5.0000, p2 = 7.0000, p3 = 3.0000
```

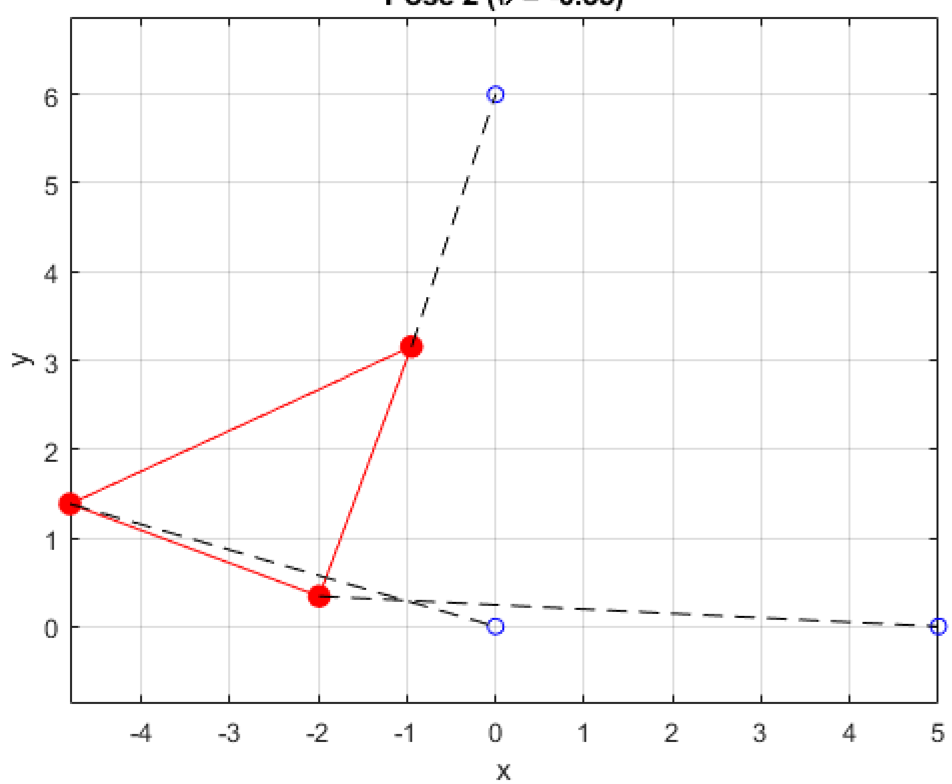

Plot of f(θ) on [-π, π] for Question #5
×10^5
2.5
2
1.5
1
0.5
0
-0.5
-4
-3
-2
-1
0
1
2
3
4
f(θ)
θ (radians)

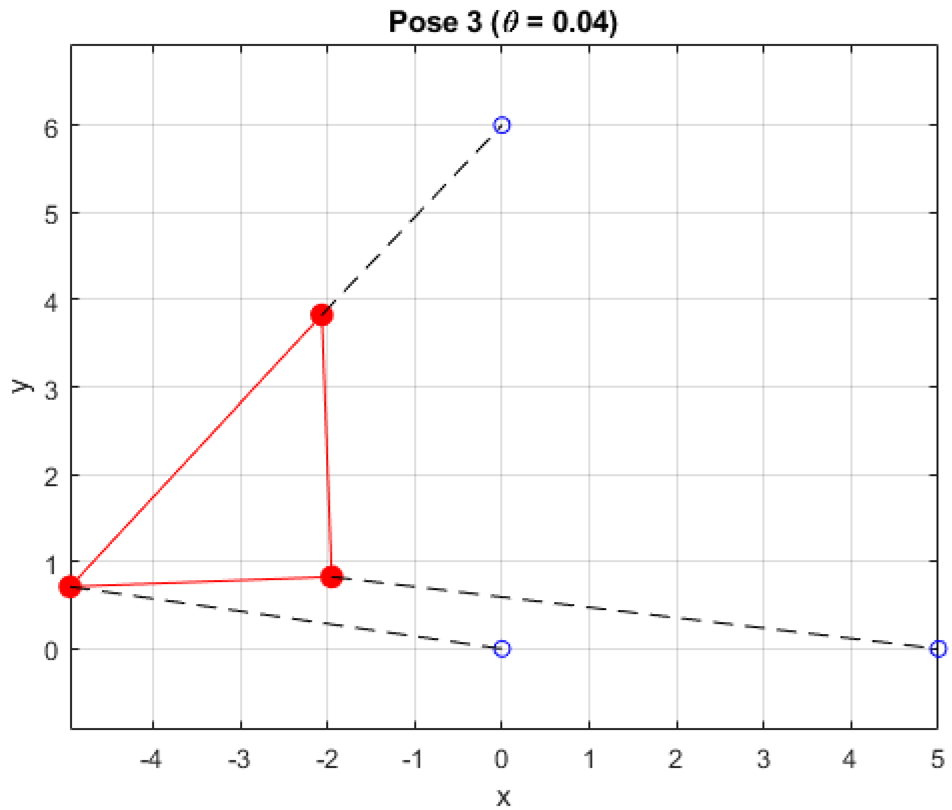

Plot of f(θ) on [-π, π] with roots
×10^5
-0.6732
-0.3547
0.0378
0.4589
0.9777
2.5139
2.5
2
1.5
1
0.5
0
-0.5
-4
-3
-2
-1
0
1
2
3
4
f(θ)
θ (radians)

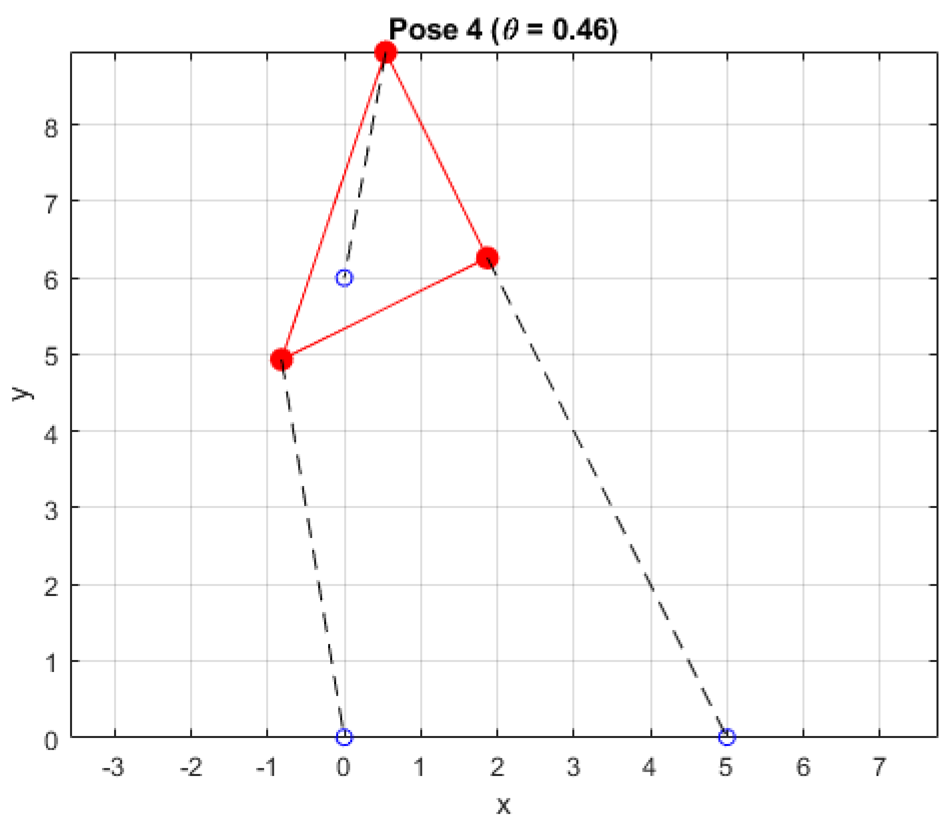
Pose 1 (θ = -0.67)

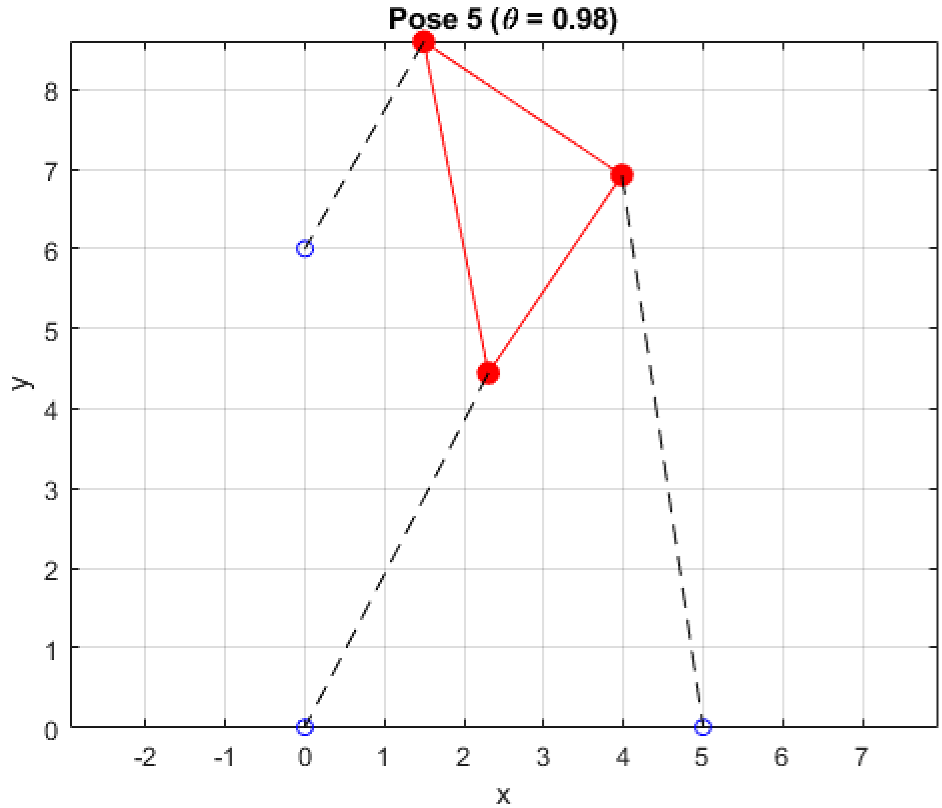


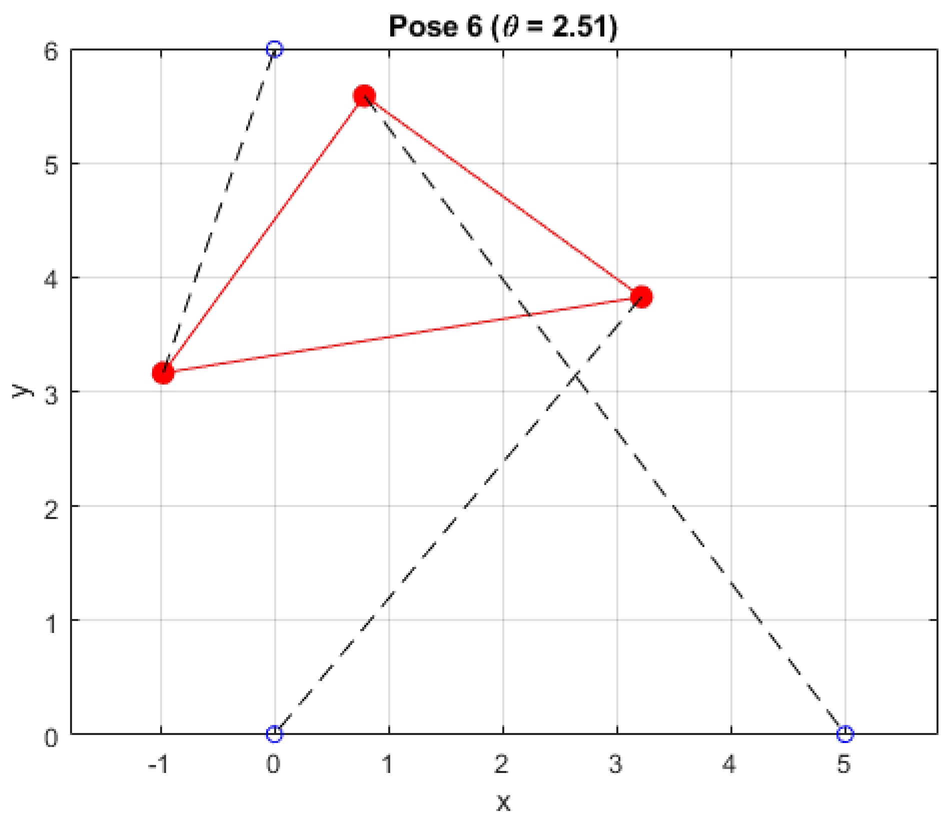
Pose 2 (θ = -0.35)




Pose 3 (θ = 0.04)
y
x
6
5
4
3
2
1
0
-4
-3
-2
-1
0
1
2
3
4
5


Pose 4 (θ = 0.46)
y
x
8
7
6
5
4
3
2
1
0
-3
-2
-1
0
1
2
3
4
5
6
7


Pose 5 (θ = 0.98)
y
x
8
7
6
5
4
3
2
1
0
-2
-1
0
1
2
3
4
5
6
7


Pose 6 (θ = 2.51)
y
x
6
5
4
3
2
1
0
-1
0
1
2
3
4
5

Published with MATLAB® R2024b