Table of Contents


```
clear;
```

QUESTION 1:

```
% Function is at the bottom in the supporting code section
% function out = f(theta)

% Testing theta = pi/4
f(pi/4)

% Testing theta = -pi/4
f(-pi/4)

% Both are close to 0, so we are good
```

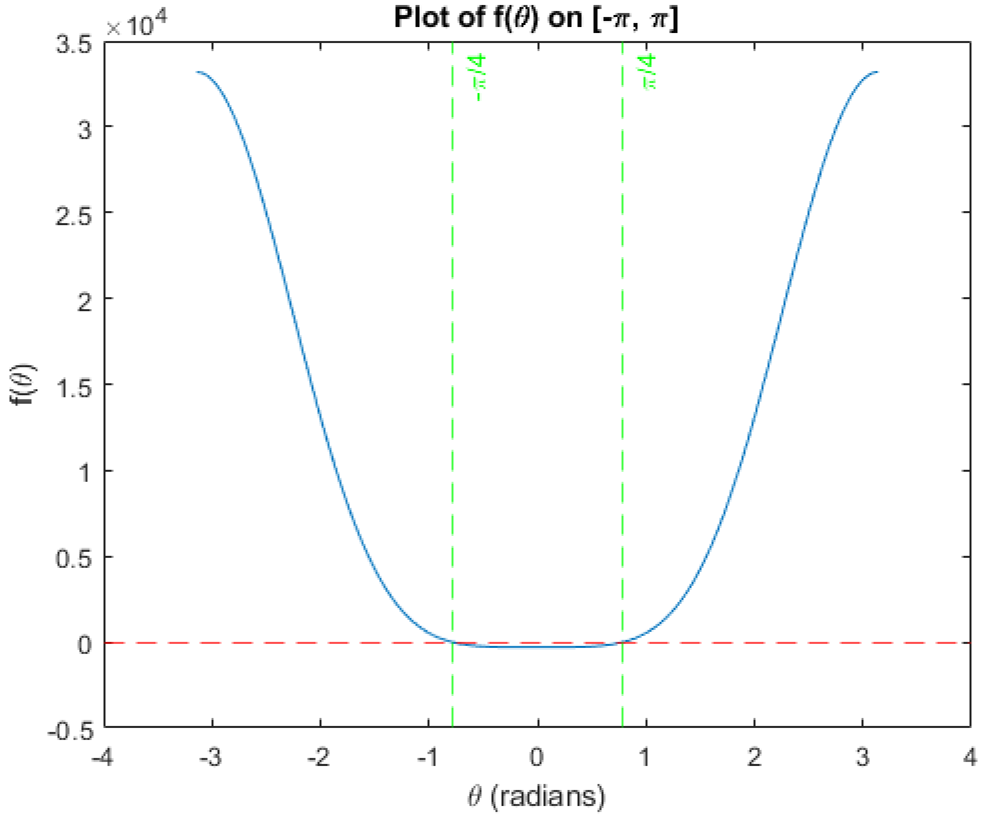
QUESTION 2:

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

f_vals = f(theta_vals);

figure(1)
plot(theta_vals, f_vals)
xlabel('\theta (radians)')
ylabel('f(\theta)')
title('Plot of f(\theta) on [-\pi, \pi]')
yline(0, '--r');
xline(pi/4, '--g', '\pi/4');
xline(-pi/4, '--g', '-\pi/4');
drawnow;

% Plot clearly shows that there are roots at +/- pi/4
```



QUESTION 3:

```
% Pose from Figure 1.15 (a)

% Connected to (0, 0) aka (x, y)
u1 = 1; v1 = 2;

% Connected to (x1, 0)
u2 = 2; v2 = 1;

% Connected to (x2, y2)
u3 = 2; v3 = 3;

x1 = 4; x2 = 0; y2 = 4;

figure(2)
plot([u1 u2 u3 u1], [v1 v2 v3 v1], 'r'); hold on          % Platform triangle
plot([0 x1 x2], [0 0 y2], 'bo')                          % Base anchors
plot([u1 u2 u3], [v1 v2 v3], 'ro', 'MarkerSize', 8, 'MarkerFaceColor', 'r')
% Platform joints
plot([u1 0], [v1 0], 'k--')     % p1
plot([u2 x1], [v2 0], 'k--')    % p2
plot([u3 x2], [v3 y2], 'k--')  % p3
title('Figure 1.15 (a)')
xlabel('x')
```

```
ylabel('y')
drawnow;

% Pose from Figure 1.15 (b)

% Connected to (0, 0) aka (x, y)
u1 = 2; v1 = 1;

% Connected to (x1, 0)
u2 = 3; v2 = 2;

% Connected to (x2, y2)
u3 = 1; v3 = 2;

x1 = 4; x2 = 0; y2 = 4;

figure(3)
plot([u1 u2 u3 u1], [v1 v2 v3 v1], 'r'); hold on          % Platform triangle
plot([0 x1 x2], [0 0 y2], 'bo')                            % Base anchors
plot([u1 u2 u3], [v1 v2 v3], 'ro', 'MarkerSize', 8, 'MarkerFaceColor', 'r')
% Platform joints
plot([u1 0], [v1 0], 'k--')    % p1
plot([u2 x1], [v2 0], 'k--')   % p2
plot([u3 x2], [v3 y2], 'k--')  % p3
title('Figure 1.15 (b)')
xlabel('x')
ylabel('y')
% drawnow;

% Here, we're just reproducing Figure 1.15 (a) and (b)
```

Figure 1.15 (a)

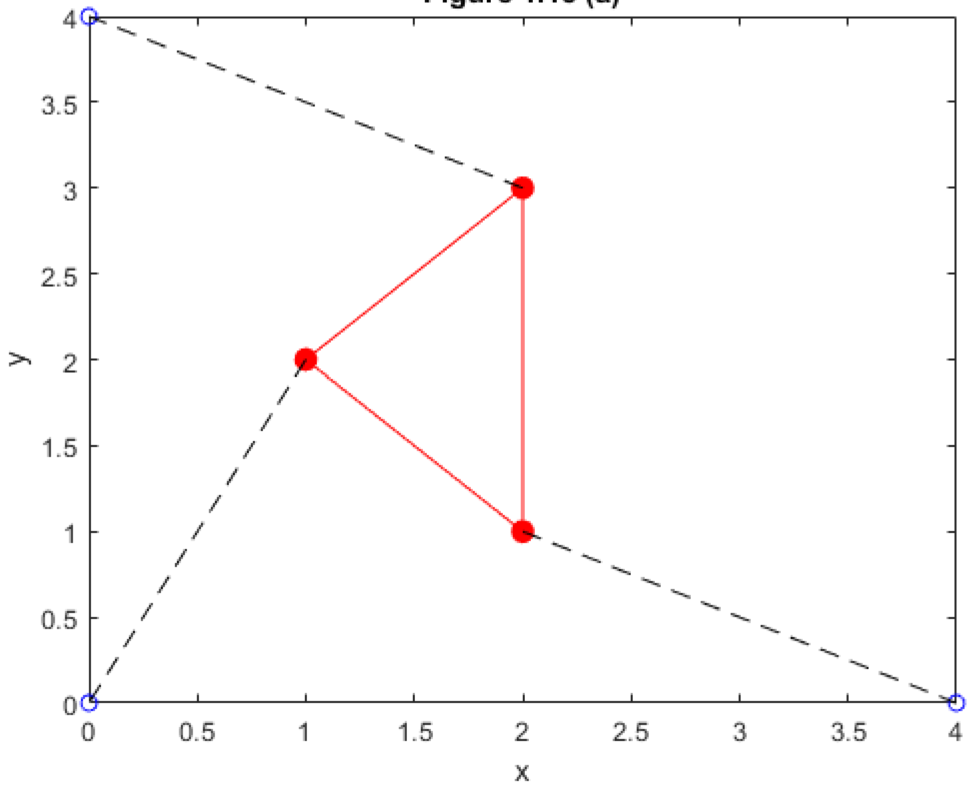
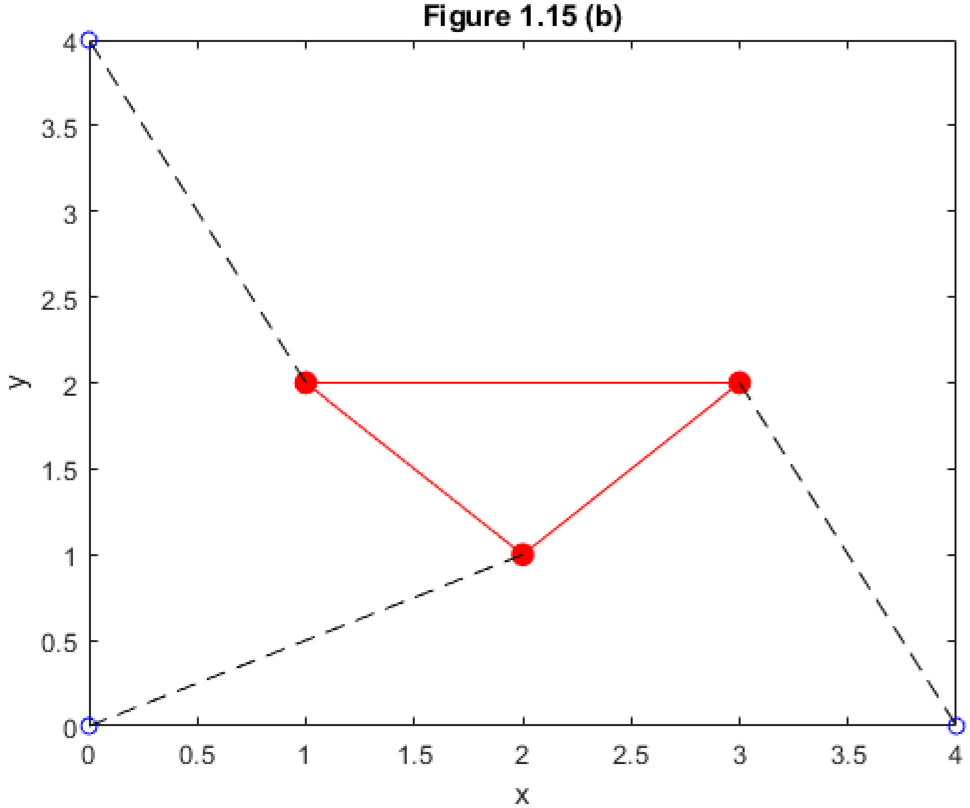


Figure 1.15 (b)



QUESTION 4:

```
% Forward kinematics is when we compute (x, y) and theta for each given p1,
% p2, and p3

% The inverse kinematic problem is when we find p1, p2, p3, given x, y, and
% theta

% The new f(theta) function is in the supporting functions section at the
% bottom (f_4(theta))

% Plotting f_4(theta) on [-pi, pi]
theta_vals = -pi:0.01:pi;

f_vals = f_4(theta_vals);

figure(4)
plot(theta_vals, f_vals)
xlabel('\theta (radians)')
ylabel('f(\theta)')
title('Plot of f(\theta) on [-\pi, \pi]')
yline(0, '--r');
drawnow;

% Finding the four theta values (guesses are from eyeballing the graph)
theta1 = fzero(@f_4, -0.72);
theta2 = fzero(@f_4, -0.33);
theta3 = fzero(@f_4, 1.14);
theta4 = fzero(@f_4, 2.11);
thetas = [theta1 theta2 theta3 theta4];

% From the above it appears that our roots are at:
% theta = -0.7208, -0.3310, 1.1437, and 2.1159 radians
figure(5)
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', '-0.7208');
xline(theta2, '--r', '-0.3310');
xline(theta3, '--r', '1.1437');
xline(theta4, '--r', '2.1159');
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
% Created a new function at the bottom called forward_kinematics
[x_1 y_1] = forward_kinematics(theta1);
[x_2 y_2] = forward_kinematics(theta2);
[x_3 y_3] = forward_kinematics(theta3);
```

```
[x_4 y_4] = forward_kinematics(theta4);
xs = [x_1 x_2 x_3 x_4];
ys = [y_1 y_2 y_3 y_4];

% It was found that
% (x_1, y_1) = (-1.3784, 4.8063)
% (x_2, y_2) = (-0.9147, 4.9156)
% (x_3, y_3) = (4.4818, 2.2167)
% (x_4, y_4) = (4.5718, 2.0244)

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

gamma = pi/4;
x1 = 5; x2 = 0; y2 = 6;

for i = 1:4
    draw_pose_4(5+i, xs(i), ys(i), thetas(i));
    drawnow;

    % Compute triangle corners again (same as inside draw_pose_4)
    x = xs(i); y = ys(i); theta = thetas(i);

    % This one is connected to (0,0)
    u1 = x;
    v1 = y;

    % This one is connected to (x1, 0)
    u2 = x + 3*cos(theta);
    v2 = y + 3*sin(theta);

    % This one is connected to (x2, y2)
    u3 = x + 3*sqrt(2)*cos(theta + gamma);
    v3 = y + 3*sqrt(2)*sin(theta + gamma);

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

    fprintf("Pose %d: p1 = %.4f, p2 = %.4f, p3 = %.4f\n", i, p1, p2, p3);
end

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

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

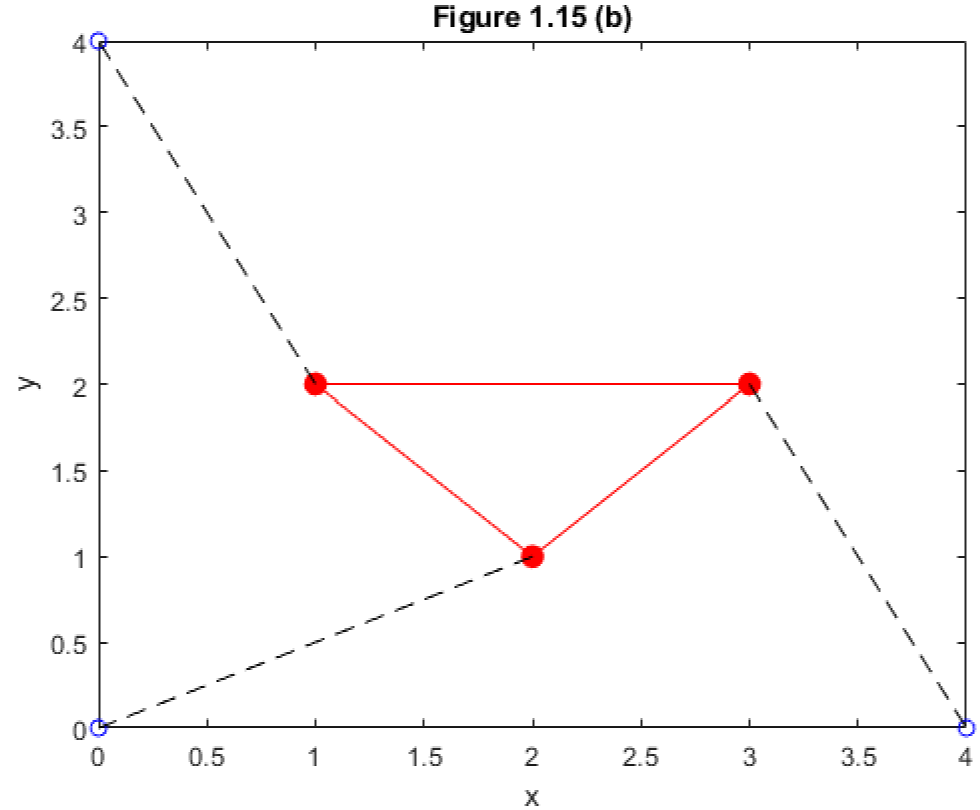
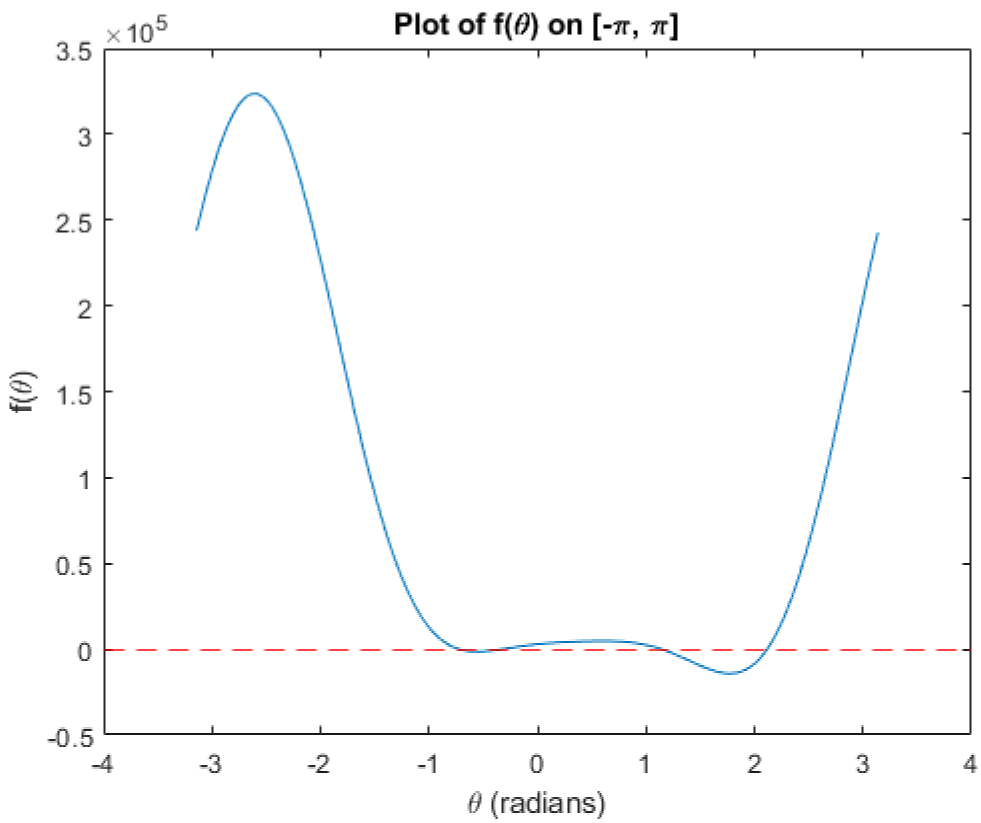
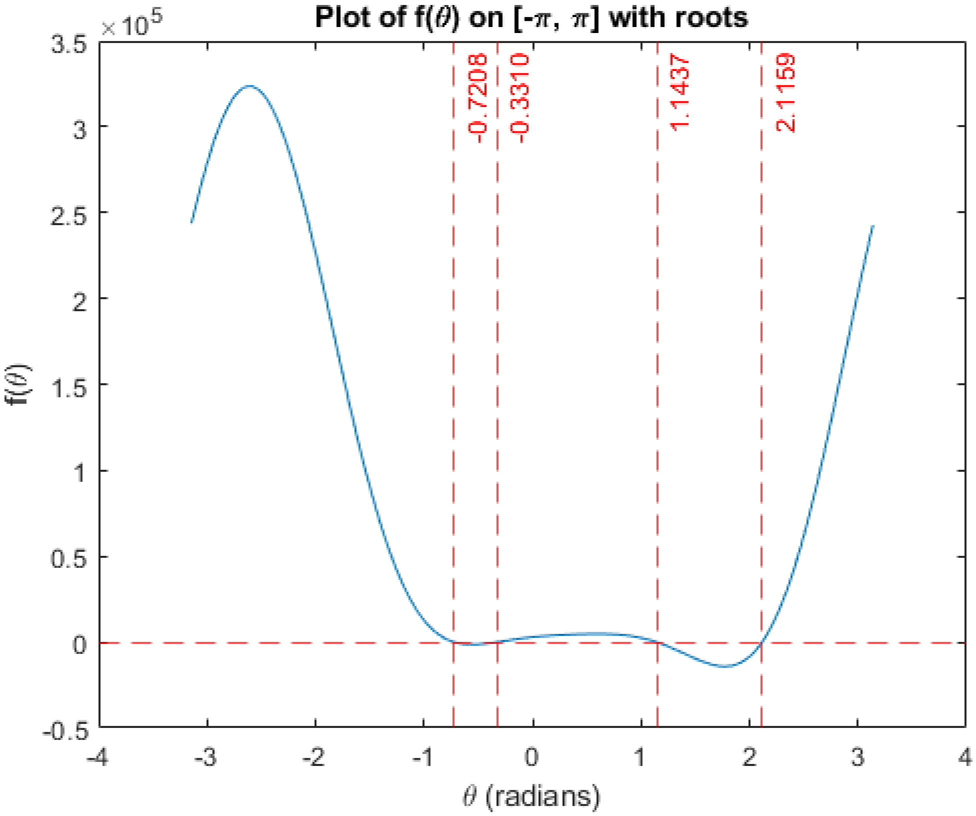
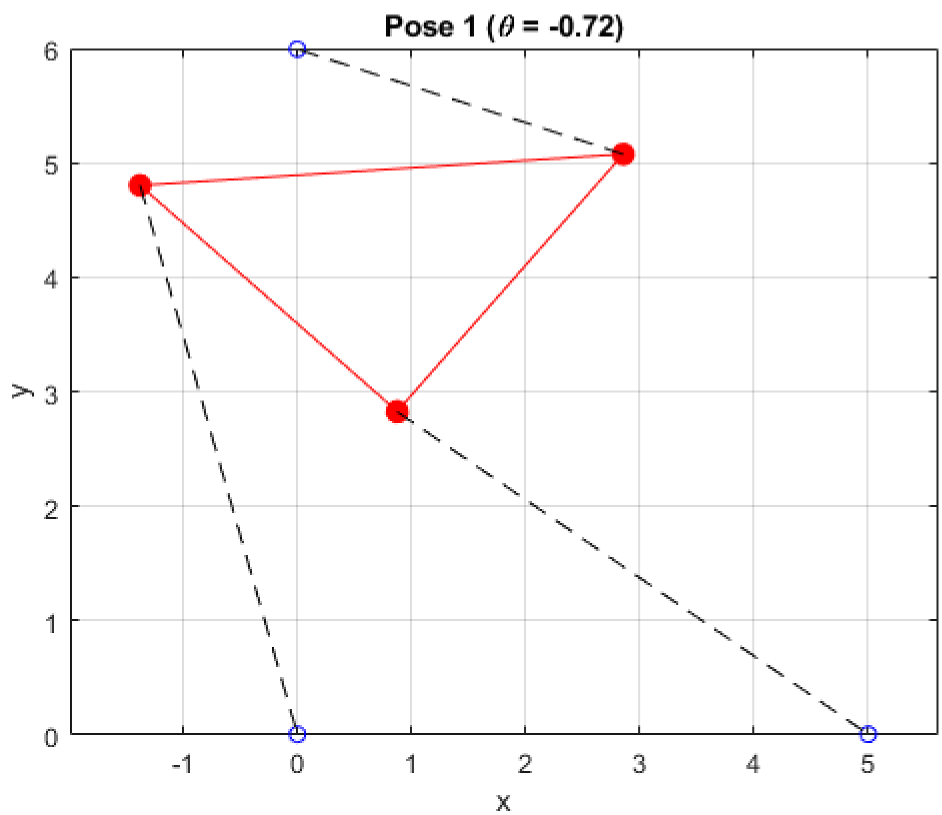


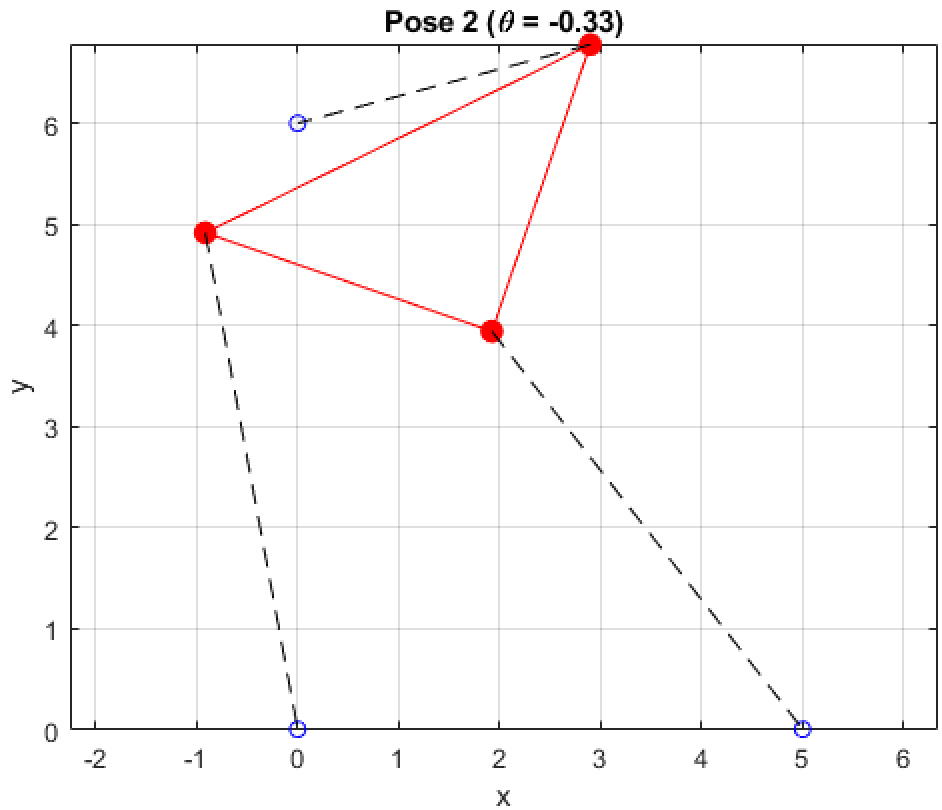
Figure 1.15 (b)
y
x
0
0.5
1
1.5
2
2.5
3
3.5
4

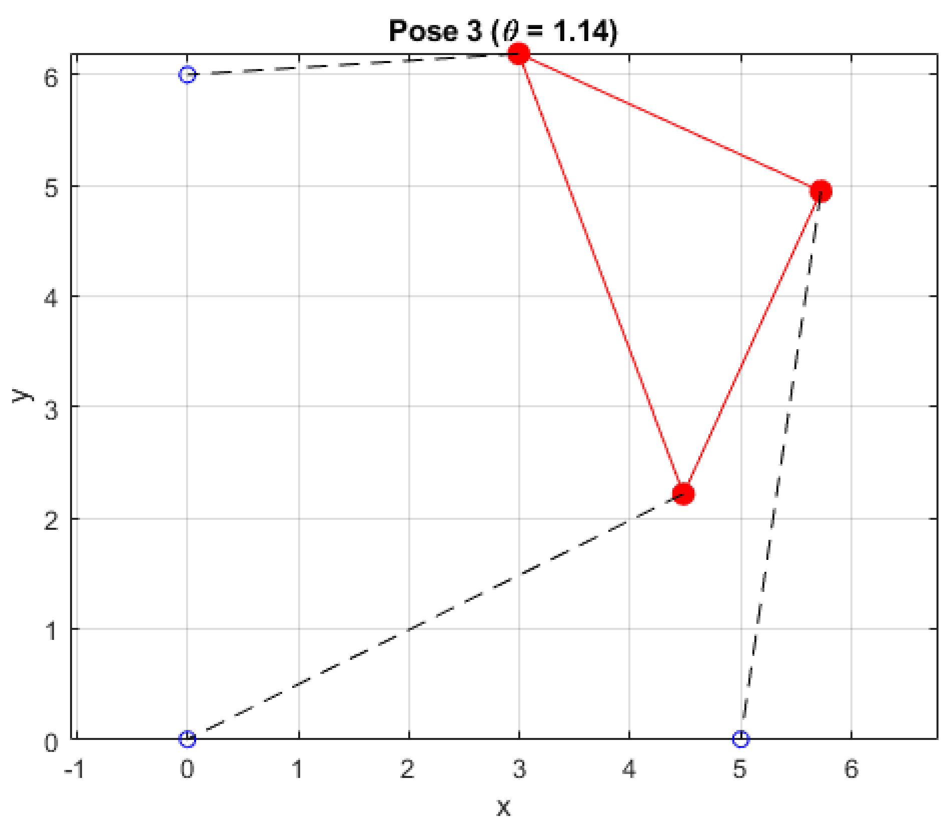


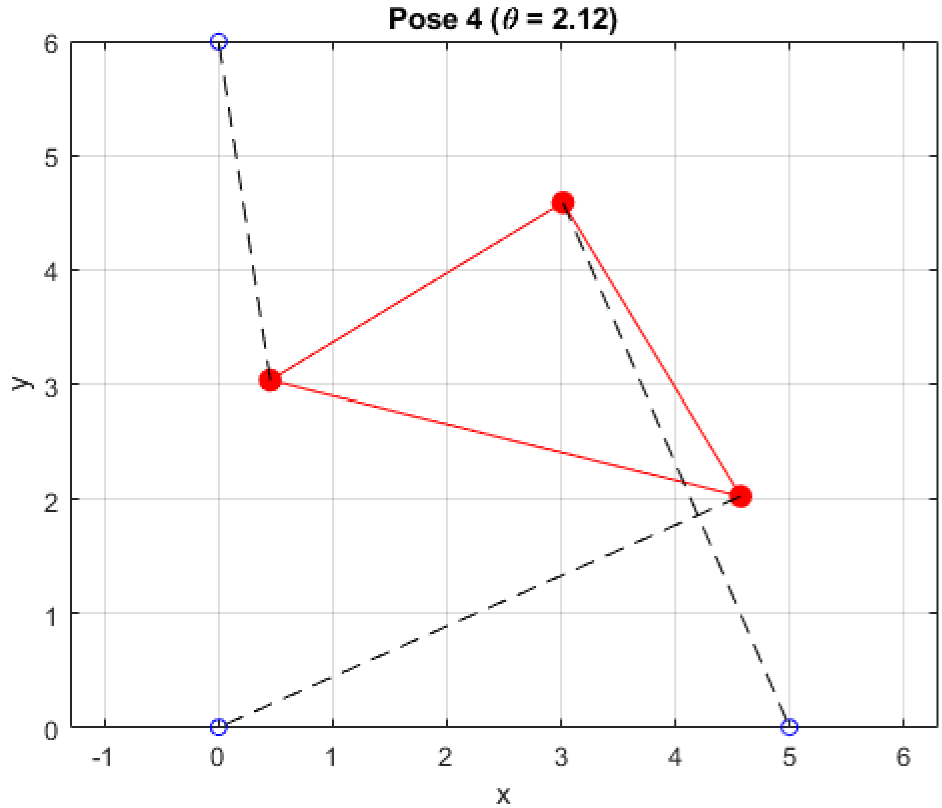
Plot of f(θ) on [-π, π]
×10^5
f(θ)
θ (radians)
3.5
3
2.5
2
1.5
1
0.5
0
-0.5
-4
-3
-2
-1
0
1
2
3
4


Plot of f(θ) on [-π, π] with roots
×10^5
-0.7208
-0.3310
1.1437
2.1159
f(θ)
θ (radians)


Pose 1 (θ = -0.72)
y
x


Pose 2 (θ = -0.33)
y
x


Pose 3 (θ = 1.14)
y
x



QUESTION 5:

QUESTION 6:

QUESTION 7:

QUESTION 8:

ALL FUNCTIONS SUPPORTING THIS CODE

```
% First f(theta) function
function out = f(theta)

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

    % Angle across from L1
    gamma = pi / 2;

    % Strut lengths
```

```
    p1 = sqrt(5);
    p2 = sqrt(5);
    p3 = sqrt(5);

    % Strut base positions
    % Got these from Figure 1.15
    x1 = 4;
    x2 = 0;
    y2 = 4;

    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

% f(theta) function for part 4
function out = f_4(theta)

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

    % Angle across from L1
    gamma = pi / 4;

    % Strut lengths
    p1 = 5;
    p2 = 5;
    p3 = 3;

    % Strut base positions
    % Got these from Figure 1.15
    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
function [x y] = forward_kinematics(theta)

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

    % Angle across from L1
    gamma = pi / 4;

    % Strut lengths
    p1 = 5;
    p2 = 5;
    p3 = 3;

    % Strut base positions
    % Got these from Figure 1.15
    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);

    x = N1 / D;
    y = N2 / D;

end

% Helper function to draw the poses for question four
function draw_pose_4(fig_num, x, y, theta)

    L2 = 3 * sqrt(2);
    L3 = 3;
```

```
    gamma = pi/4;

    % This one is connected to (0,0)
    u1 = x;
    v1 = y;

    % This one is connected to (x1, 0)
    u2 = x + L3*cos(theta);
    v2 = y + L3*sin(theta);

    % This one is connected to (x2, y2)
    u3 = x + L2*cos(theta + gamma);
    v3 = y + L2*sin(theta + gamma);

    x1 = 5; x2 = 0; y2 = 6;

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

    new_num = fig_num - 5;
    title(sprintf('Pose %d (\\theta = %.2f)', new_num, theta))
    xlabel('x')
    ylabel('y')
    axis equal
    grid on
end
```

```
ans =

  -4.5475e-13


ans =

  -4.5475e-13
```

Published with MATLAB® R2024b