

---

## Table of Contents

QUESTION 8: .....	1
ALL FUNCTIONS SUPPORTING .....	2

## QUESTION 8:

```
% 1. Define base attachment points (fixed)
B = [... % Each row is a point (xi, yi, zi)
      200, 0, 0;
      100, 173.2, 0;
      -100, 173.2, 0;
      -200, 0, 0;
      -100, -173.2, 0;
      100, -173.2, 0
];

% 2. Define platform attachment points (in platform frame)
P = [... % Each row is a point (xi, yi, zi)
      100, 0, 0;
      50, 86.6, 0;
      -50, 86.6, 0;
      -100, 0, 0;
      -50, -86.6, 0;
      50, -86.6, 0
];

% 3. Define leg lengths (example values)
L = [250; 240; 245; 260; 255; 250]; % Length of each leg

% 4. Objective function to minimize
f_obj = @(x) stewart_error(x, B, P, L);

% 5. Initial guess for pose: [x, y, z, alpha, beta, gamma]
x0 = [0; 0; 200; 0; 0; 0]; % Start near z = 200 mm, no rotation

% 6. Use fsolve to solve
options = optimoptions('fsolve', 'Display', 'iter', 'TolFun', 1e-10);
[x_sol, fval, exitflag] = fsolve(f_obj, x0, options);

disp('Estimated pose:')
disp(['x = ', num2str(x_sol(1)), ', y = ', num2str(x_sol(2)), ...
      ', z = ', num2str(x_sol(3))])
disp(['alpha = ', num2str(x_sol(4)), ', beta = ', num2str(x_sol(5)), ...
      ', gamma = ', num2str(x_sol(6))])

interactive_stewart(B, P);
```

---

# ALL FUNCTIONS SUPPORTING

```
function F = stewart_error(x, B, P, L)
    % Inputs:
    % x = [x; y; z; alpha; beta; gamma]

    pos = x(1:3);           % Translation vector
    alpha = x(4); beta = x(5); gamma = x(6); % Roll, pitch, yaw

    % Rotation matrix (ZYX order)
    Rz = [cos(gamma) -sin(gamma) 0;
          sin(gamma)  cos(gamma) 0;
          0 0 1];
    Ry = [cos(beta) 0 sin(beta);
          0 1 0;
          -sin(beta) 0 cos(beta)];
    Rx = [1 0 0;
          0 cos(alpha) -sin(alpha);
          0 sin(alpha) cos(alpha)];

    R = Rz * Ry * Rx;

    F = zeros(6,1);
    for i = 1:6
        L_vec = pos + R * P(i,:) - B(i,:);
        F(i) = norm(L_vec)^2 - L(i)^2; % Constraint: length match
    end
end

function visualize_stewart(B, P, pose, ax)
    if nargin < 4
        figure;
        ax = gca;
    end

    % Extract pose
    pos = pose(1:3);
    alpha = pose(4);
    beta = pose(5);
    gamma = pose(6);

    % Rotation matrix
    Rz = [cos(gamma), -sin(gamma), 0;
          sin(gamma),  cos(gamma), 0;
          0,          0,          1];
    Ry = [cos(beta), 0, sin(beta);
          0,          1, 0;
          -sin(beta), 0, cos(beta)];
    Rx = [1, 0, 0;
          0, cos(alpha), -sin(alpha);
          0, sin(alpha),  cos(alpha)];
    R = Rz * Ry * Rx;
```

---

---

```

% Transform
P_world = (R * P')' + pos';
B_loop = [B; B(1,:)];
P_loop = [P_world; P_world(1,:)];

% --- Plot in the given axes ---
cla(ax);
axes(ax); %#ok<LAXES>
hold on; grid on; axis equal;
xlabel('X'); ylabel('Y'); zlabel('Z');
title('Stewart Platform Visualization');
view(3);

% Plot
plot3(ax, B_loop(:,1), B_loop(:,2), B_loop(:,3), 'bo-', 'LineWidth', 2);
plot3(ax, P_loop(:,1), P_loop(:,2), P_loop(:,3), 'ro-', 'LineWidth', 2);
for i = 1:6
    plot3(ax, [B(i,1), P_world(i,1)], ...
           [B(i,2), P_world(i,2)], ...
           [B(i,3), P_world(i,3)], 'k--', 'LineWidth', 1.5);
end

legend(ax, 'Base', 'Platform', 'Struts')
end

function interactive_stewart(B, P)
figure('Name', 'Interactive Stewart Platform');

% Initial pose
pose = [0; 0; 200; 0; 0; 0]; % [x y z alpha beta gamma]

% Create sliders for each parameter
labels = {'X', 'Y', 'Z', 'Roll ( $\alpha$ )', 'Pitch ( $\beta$ )', 'Yaw ( $\gamma$ )'};
mins    = [-100, -100, 150, -pi, -pi, -pi];
maxs    = [ 100,  100, 300,  pi,  pi,  pi];

sliders = gobjects(1,6);
for i = 1:6
    uicontrol('Style', 'text', 'String', labels{i}, ...
              'Position', [20, 400 - 50*i, 80, 20]);
    sliders(i) = uicontrol('Style', 'slider', ...
                          'Min', mins(i), 'Max', maxs(i), ...
                          'Value', pose(i), ...
                          'Position', [100, 400 - 50*i, 300, 20], ...
                          'Callback', @(src, ~) update_plot());
end

% Axes for visualization
ax = axes('Position', [0.5 0.2 0.45 0.7]);
view(3); grid on; axis equal;
xlabel('X'); ylabel('Y'); zlabel('Z');
title('Stewart Platform Pose');

```

---

---

```

% Initial plot
update_plot();

% Update function
function update_plot()
    for i = 1:6
        pose(i) = sliders(i).Value;
    end
    cla(ax);
    axes(ax);
    visualize_stewart(B, P, pose, ax);
end
end

```

			Norm of	First-order
Trust-region				
Iteration	Func-count	$  f(x)  ^2$	step	
optimality	radius			
0	7	1.00631e+09		
3.43e+08	1			
1	14	9.48743e+08	0.0959583	
1.03e+08	1			
2	21	9.12547e+08	1	
2.89e+08	1			
3	22	9.12547e+08	2.5	
2.89e+08	2.5			
4	29	7.65115e+07	0.625	
3.88e+08	0.625			
5	36	1.81799e+07	1.5625	
2.3e+07	1.56			
6	43	1.80569e+07	3.90625	
9.08e+06	3.91			
7	44	1.80569e+07	3.90625	
9.08e+06	3.91			
8	51	1.80381e+07	0.976562	
5.86e+05	0.977			
9	52	1.80381e+07	2.44141	
5.86e+05	2.44			
10	53	1.80381e+07	0.610352	
5.86e+05	0.61			
11	60	1.8038e+07	0.152588	
9.78e+04	0.153			
12	67	1.8038e+07	0.152588	
3.34e+05	0.153			
13	74	1.80379e+07	0.152588	
9.78e+04	0.153			
14	81	1.80379e+07	0.152588	
2.19e+05	0.153			
15	88	1.80379e+07	0.152588	
9.78e+04	0.153			
16	95	1.80379e+07	0.152588	
1.4e+05	0.153			
17	96	1.80379e+07	0.152588	

---

---

1.4e+05	0.153		
18	103	1.80379e+07	0.038147
9.75e+04	0.0381		
19	110	1.80379e+07	0.038147
1.27e+05	0.0381		
20	117	1.80379e+07	0.0953674
1.01e+05	0.0954		
21	124	1.80379e+07	0.0953674
1.05e+05	0.0954		
22	131	1.80379e+07	0.0953674
9.71e+04	0.0954		
23	138	1.80379e+07	0.0953674
9.86e+04	0.0954		
24	145	1.80379e+07	0.0953674
9.7e+04	0.0954		
25	152	1.80379e+07	0.0953674
9.82e+04	0.0954		
26	159	1.80379e+07	0.0953674
9.68e+04	0.0954		
27	166	1.80379e+07	0.0953674
9.8e+04	0.0954		
28	173	1.80378e+07	0.0953674
9.67e+04	0.0954		
29	180	1.80378e+07	0.0953674
9.82e+04	0.0954		
30	187	1.80378e+07	0.0953674
9.7e+04	0.0954		
31	194	1.80378e+07	0.0953674
9.83e+04	0.0954		
32	201	1.80378e+07	0.0953674
9.68e+04	0.0954		
33	208	1.80378e+07	0.0953674
9.8e+04	0.0954		
34	215	1.80378e+07	0.0953674
9.67e+04	0.0954		
35	222	1.80378e+07	0.0953674
9.81e+04	0.0954		
36	229	1.80378e+07	0.0953674
9.67e+04	0.0954		
37	236	1.80378e+07	0.0953674
9.81e+04	0.0954		
38	243	1.80378e+07	0.0953674
9.67e+04	0.0954		
39	250	1.80378e+07	0.0953674
9.91e+04	0.0954		
40	257	1.80378e+07	0.0953674
9.65e+04	0.0954		
41	264	1.80378e+07	0.0953674
9.79e+04	0.0954		
42	271	1.80377e+07	0.0953674
9.65e+04	0.0954		
43	278	1.80377e+07	0.0953674
1.04e+05	0.0954		
44	285	1.80377e+07	0.0953674

---

---

9.64e+04	0.0954		
45	292	1.80377e+07	0.0953674
9.79e+04	0.0954		
46	299	1.80377e+07	0.0953674
9.64e+04	0.0954		
47	306	1.80377e+07	0.0953674
9.77e+04	0.0954		
48	313	1.80377e+07	0.0953674
9.64e+04	0.0954		
49	320	1.80377e+07	0.0953674
9.77e+04	0.0954		
50	327	1.80377e+07	0.0953674
9.64e+04	0.0954		
51	334	1.80377e+07	0.0953674
9.79e+04	0.0954		
52	341	1.80377e+07	0.0953674
9.64e+04	0.0954		
53	348	1.80377e+07	0.0953674
9.76e+04	0.0954		
54	355	1.80377e+07	0.0953674
9.62e+04	0.0954		
55	362	1.80377e+07	0.0953674
9.74e+04	0.0954		
56	369	1.80377e+07	0.0953674
9.61e+04	0.0954		
57	376	1.80377e+07	0.0953674
9.74e+04	0.0954		
58	383	1.80377e+07	0.0953674
9.6e+04	0.0954		
59	390	1.80376e+07	0.0953674
9.74e+04	0.0954		
60	397	1.80376e+07	0.0953674
9.62e+04	0.0954		
61	404	1.80376e+07	0.0953674
1.02e+05	0.0954		
62	411	1.80376e+07	0.0953674
9.64e+04	0.0954		
63	418	1.80376e+07	0.0953674
1.08e+05	0.0954		
64	425	1.80376e+07	0.0953674
9.65e+04	0.0954		
65	432	1.80376e+07	0.0953674
9.88e+04	0.0954		
66	439	1.80376e+07	0.0953674
9.64e+04	0.0954		
67	446	1.80376e+07	0.0953674
9.76e+04	0.0954		
68	453	1.80376e+07	0.0953674
9.64e+04	0.0954		
69	460	1.80376e+07	0.0953674
9.77e+04	0.0954		
70	467	1.80376e+07	0.0953674
9.64e+04	0.0954		
71	474	1.80376e+07	0.0953674

---

---

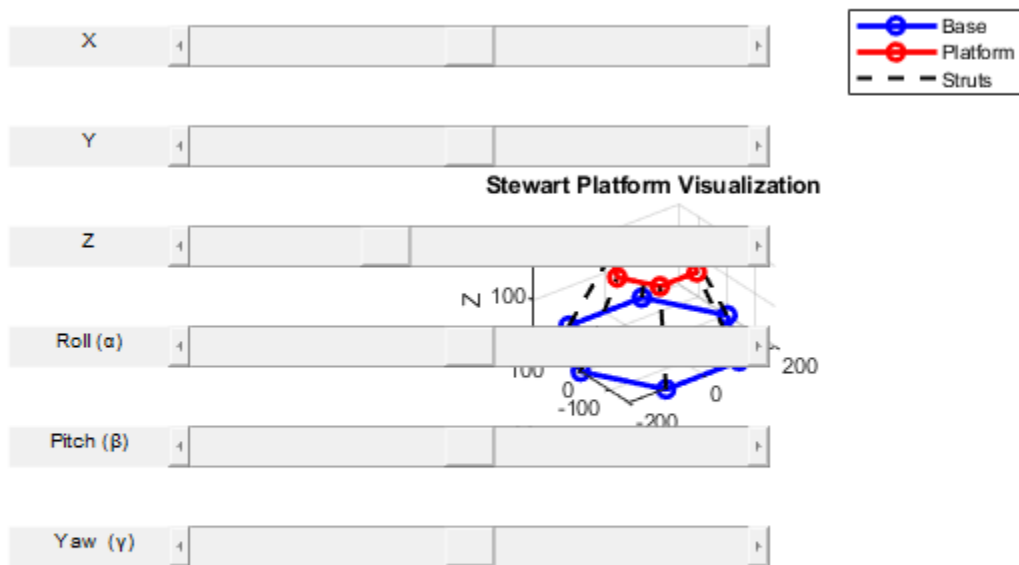
9.79e+04	0.0954		
72	481	1.80376e+07	0.0953674
9.66e+04	0.0954		
73	488	1.80376e+07	0.0953674
9.81e+04	0.0954		
74	495	1.80375e+07	0.0953674
9.66e+04	0.0954		
75	502	1.80375e+07	0.0953674
9.78e+04	0.0954		
76	509	1.80375e+07	0.0953674
9.64e+04	0.0954		
77	516	1.80375e+07	0.0953674
9.76e+04	0.0954		
78	523	1.80375e+07	0.0953674
9.63e+04	0.0954		
79	530	1.80375e+07	0.0953674
9.76e+04	0.0954		
80	537	1.80375e+07	0.0953674
9.61e+04	0.0954		
81	544	1.80375e+07	0.0953674
9.74e+04	0.0954		
82	551	1.80375e+07	0.0953674
9.59e+04	0.0954		
83	558	1.80375e+07	0.0953674
9.73e+04	0.0954		
84	565	1.80375e+07	0.0953674
9.62e+04	0.0954		
85	572	1.80375e+07	0.0953674
9.74e+04	0.0954		
86	579	1.80375e+07	0.0953674
9.62e+04	0.0954		
87	586	1.80375e+07	0.0953674
9.76e+04	0.0954		
88	593	1.80375e+07	0.0953674
9.64e+04	0.0954		
89	600	1.80374e+07	0.0953674
9.77e+04	0.0954		

*Solver stopped prematurely.*

*fsolve stopped because it exceeded the function evaluation limit,  
options.MaxFunctionEvaluations = 6.000000e+02.*

*Estimated pose:*

*x = 0.052786, y = -0.99558, z = 205.3665  
alpha = -0.075693, beta = 0.067383, gamma = -0.73633*



*Published with MATLAB® R2024b*