Contents

- Question1 3D workspace optional , side view and top view are there in the PDF (Hand Drawn)
- Question 2 Co-ordinate System drawing using DH parameters
- Question 3 DH paramter's table
- Question 4 The Composite Transformation
- Position and Orientation Forward kinematics by solving composite Transformation
- Question 5 Forward kinematics for home position for values given ie all joint angles are zero and the link lengths are l1=0 , l2=70 and l3= 100
- Question 6
- Question 7 for inverse kinematics
- Question 8 Jacobian of the leg
- Question 9 Singularities of the leg please read the comment
- Question 10 Joint velocity at home position i.e. all joint angles are zero
- Function for forward kinematics


```
close all
clc
theta1= sym('theta1','real');
theta2= sym('theta2','real');
theta3= sym('theta3','real');
l1= sym('l1','real');
l2= sym('l2','real');
l3= sym('l3','real');
pi = sym('pi');
syms r11 r12 r13 r21 r22 r23 r31 r32 r33
deg2rad=pi/180;
link_number = [1,2,3,4]';
theta = [theta1,theta2,theta3,pi/2]';
d = [0,0,l3,0]';
alpha = [pi/2,0,pi/2,0]';
a = [l1,l2,0,0]';
```

Question1 3D workspace optional , side view and top view are there in the PDF (Hand Drawn)

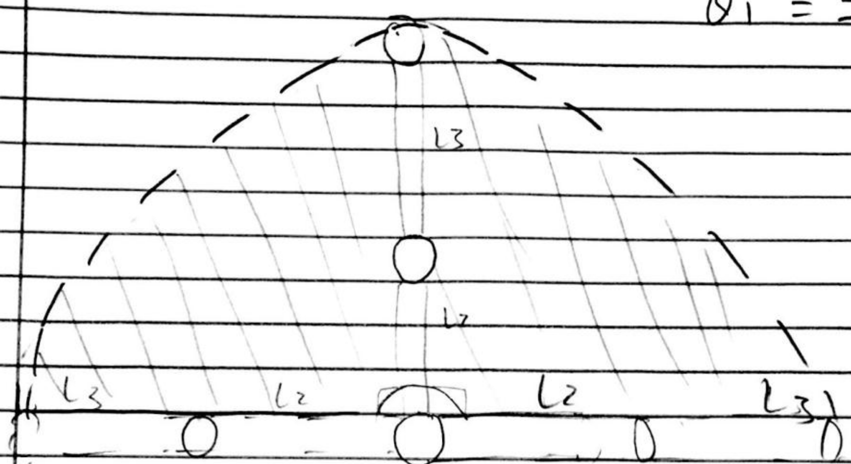
```
p = [];
index = 0;
for theta1 = -pi/4:0.1:pi/4
    for theta2 = -pi/2:0.1:pi/2
        for theta3 = -pi/3:0.1:pi/3
            index = index +1;
            x = 70*cos(theta1)*cos(theta2) - 100*sin(theta1);
            y = 100*cos(theta1) + 70*cos(theta2)*sin(theta1);
            z = -70*sin(theta2);
            p(index,:) = [x,y,z];
        end
     end
end
plot3(p(:,1),p(:,2),p(:,3),'*b')
```

Workspace.

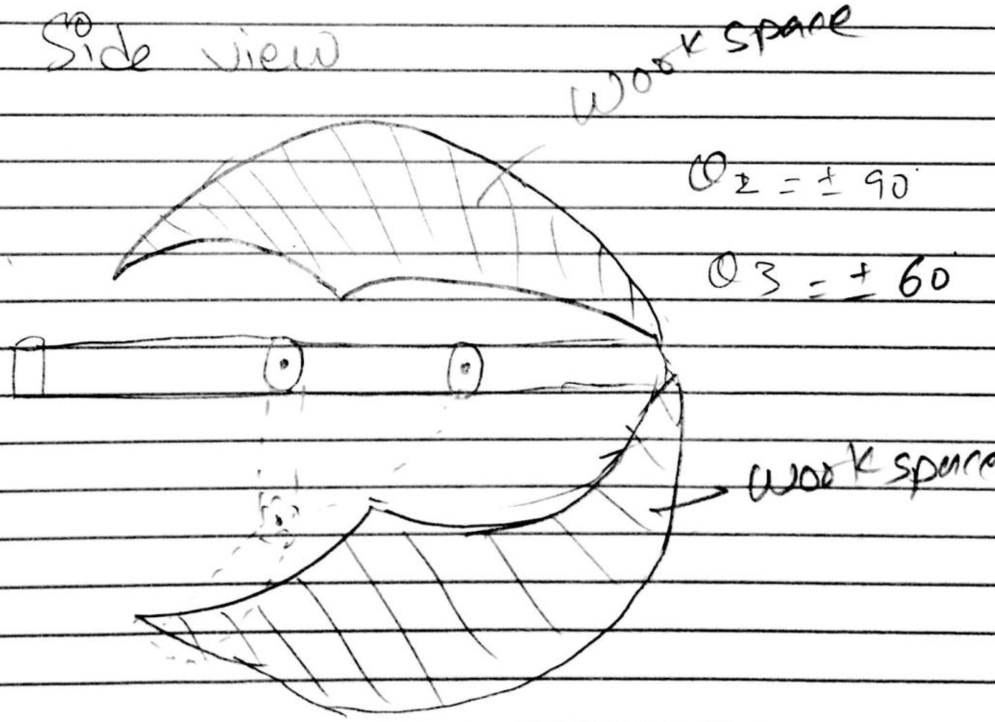
→ Top view :-

Assuming

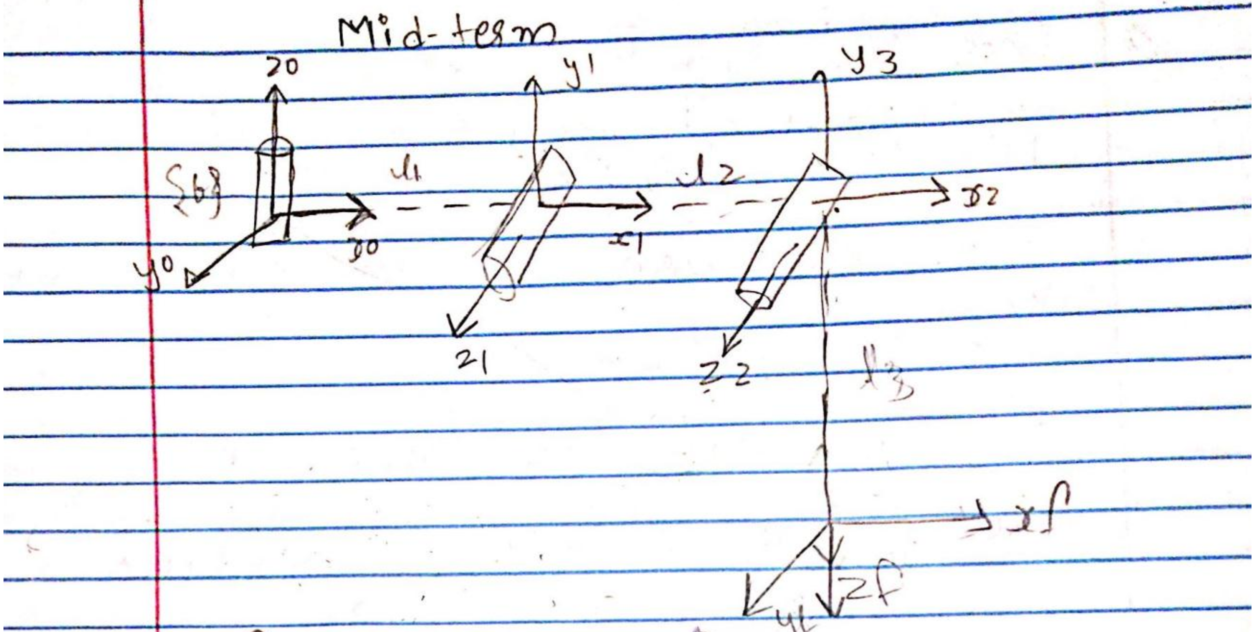
$\theta_1 = \pm 90°$



→ Side view



Question 2 Co-ordinate System drawing using DH parameters



DH parameter

link	theta	d	a	alpha
1	θ_1	0	l_1	+pi/2
2	θ_2	0	l_2	0
3	θ_3	l_3	0	+pi/2
4	pi/2	0	0	0

Question 3 DH paramter's table

```
DHParameters = [link_number,theta,d,alpha,a];
```

```
disp(DHParameters)



for n=1:3
    T(:,:,n) = Fk(theta(n),d(n),alpha(n),a(n));
end
T01 = T(:,:,1);
T12 = T(:,:,2);
T23 = T(:,:,3);
```

```
[ 1, theta1,  0, pi/2, l1]

[ 2, theta2,  0,    0, l2]

[ 3, theta3, l3, pi/2,  0]

[ 4,   pi/2,  0,    0,  0]
```

Question 4 The Composite Transformation

```
TT(:,:,1) = T(:,:,1);

for n=2:3
TT(:,:,n) = TT(:,:,n-1)*T(:,:,n);
end
T03 = simplify(TT(:,:,3))
T02 = simplify(TT(:,:,2));
```

```
T03 =



[ cos(theta2 + theta3)*cos(theta1),  sin(theta1), sin(theta2 + theta3)*cos(theta1),
l1*cos(theta1) + l3*sin(theta1) + l2*cos(theta1)*cos(theta2)]

[ cos(theta2 + theta3)*sin(theta1), -cos(theta1), sin(theta2 + theta3)*sin(theta1),
l1*sin(theta1) - l3*cos(theta1) + l2*cos(theta2)*sin(theta1)]

[              sin(theta2 + theta3),            0,             -cos(theta2 + theta3),
l2*sin(theta2)]

[                                 0,            0,                                 0,
1]
```

Position and Orientation Forward kinematics by solving composite Transformation

```
TipPositionT03 = subs(T03(1:3,4),[l1,l2,l3],[0,70,100])

%  OrientationT03 = T03(1:3,1:3);
```

```
TipPositionT03 =


 100*sin(theta1) + 70*cos(theta1)*cos(theta2)

 70*cos(theta2)*sin(theta1) - 100*cos(theta1)

                              70*sin(theta2)
```

Question 5 Forward kinematics for home position for values given ie all joint angles are zero and the link lengths are l1=0 , l2=70 and l3= 100

```
Thome = vpa(subs(T03,[theta1,theta2,theta3,l1,l2,l3],[0,0,0,0,70,100]))

TipPositionhome = Thome(1:3,4);% all joint angles are zero
OrientationThome =Thome(1:3,1:3);
```

```
Thome =


[ 1.0,    0,    0,   70.0]

[   0, -1.0,    0, -100.0]

[   0,    0, -1.0,      0]

[   0,    0,    0,    1.0]
```

Question 6

```
given_vector_in_f = [0;0; 10; 1];

vector_in_b = Thome*given_vector_in_f
```

```
vector_in_b =


   70.0

 -100.0

  -10.0

    1.0
```

Question 7 for inverse kinematics

```
l1 = 0;

l2 = 70;

l3 = 100;

tippostion_given = [80;0;-100];

x = tippostion_given(1);

y = tippostion_given(2);

z = tippostion_given(3);

link_len = sqrt((x)^2 + (y)^2 );

r = sqrt((link_len-l1)^2 + (z)^2);

alp = atan2(z,(link_len-l1));
```

```
beta = acos(((l2)^2 +(r)^2-(l3)^2)/(2*l2*r));

s = vpa(alp + beta);

gamma = acos(((l3)^2 + (l2)^2-(r)^2)/(2*l2*l3));



theta1_ik = u % value of theta1 can also be pi
theta2_ik = real(s)% here we can different value of theta2 if we place theta1 = pi
theta3_ik = vpa(pi - gamma)
```

```
theta1_ik =

     0


theta2_ik =

-0.0071634442938652842514102303539403


theta3_ik =

1.4634474106930280948928062190293
```

Question 8 Jacobian of the leg

```
Tip = TipPositionT03;

Jv = simplify ([diff(Tip,theta1),diff(Tip,theta2),diff(Tip,theta3)]);

k = [0 ; 0 ; 1];
```

```
Jw = [k,T01(1:3,1:3)*k, T02(1:3,1:3)*k];

J = [Jv; Jw]
```

```
J =


[ 100*cos(theta1) - 70*cos(theta2)*sin(theta1), -70*cos(theta1)*sin(theta2),
0]

[ 100*sin(theta1) + 70*cos(theta1)*cos(theta2), -70*sin(theta1)*sin(theta2),
0]

[                                             0,              70*cos(theta2),
0]

[                                             0,                 sin(theta1),
sin(theta1)]

[                                             0,                -cos(theta1), -
cos(theta1)]

[                                             1,                           0,
0]
```

Question 9 Singularities of the leg please read the comment

```
singularity = det(Jv)



% Here when we select the values of joint angles , which makes the rank of
% the jacobian less than its maximum value we will be having singularities
```

```
singularity =


0
```

Question 10 Joint velocity at home position i.e. all joint angles are zero

```
x_dot = [0; 0; 10;0;0;0];

J0 = vpa(subs(J,[theta1 theta2 theta3], [0 0 0]));

theta_dot = pinv(J0) * x_dot
```

```
theta_dot =

                                  0

  0.1428571428571428571428571428571414

 -0.1428571428571428571428571428571414
```

Function for forward kinematics

```
function [ transMatrix ] =Fk(theta,d,alpha,a)
    rotOldZAxis = [cos(theta) -sin(theta) 0 0;...
    sin(theta) cos(theta) 0 0;...
    0 0 1 0;...
    0 0 0 1];
    translationOldZAxis = [1 0 0 0;...
    0 1 0 0;...
    0 0 1 d;...
    0 0 0 1];
    translationNewXAxis = [1 0 0 a;...
    0 1 0 0;...
    0 0 1 0;...
    0 0 0 1];
    rotNewXAxis = [1 0 0 0;...
    0 cos(alpha) -sin(alpha) 0;...
    0 sin(alpha) cos(alpha) 0;...
    0 0 0 1];

    transMatrix = rotOldZAxis*translationOldZAxis*translationNewXAxis*rotNewXAxis;
```

```
end
```

Published with MATLAB® R2017a