```
% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                               PROBLEM 1
% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
disp('#########################################################################')
disp('                               PROBLEM 1                                 ')
disp('#########################################################################')

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                   FORWARD KINEMATICS
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%Matlab Code
syms alpha theta a d

A(theta,d,a,alpha)= [cos(theta), -sin(theta)*cos(alpha) ...
 sin(theta)*sin(alpha), a*cos(theta);
 sin(theta),  cos(theta)*cos(alpha), ...
-cos(theta)*sin(alpha), a*sin(theta);
0 sin(alpha) cos(alpha) d;
0 0 0 1];

syms d1 t1 t2 t3 t5 t6 l1 l2 l3 l4 l5 l6 d4
 A_0_1 = simplify(A(t1+pi/2,l1,0,-pi/2));
 A_1_2 = simplify(A(t2+pi/2,0, 0, pi/2));
 A_2_3 = simplify(A(t3-pi/2, l2+l3 , -l4 ,0));
 A_3_4 = simplify(A(-pi/2, l5+d4, 0 ,-pi/2));
 A_4_5 = simplify(A(t5, l4, 0, pi/2));
 A_5_n = simplify(A(t6, l6, 0, 0));

 T_0_1 = A_0_1;
 T_0_2 = T_0_1*A_1_2;
 T_0_3 = T_0_2*A_2_3;
 T_0_4 = T_0_3*A_3_4;
 T_0_5 = T_0_4*A_4_5;
 T_0_n = simplify(T_0_5*A_5_n);
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%       Testing Configuration
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
disp('########################################################################')
disp('       T_0_n formula (EndEffector frame wrt to base frame      ')
disp('########################################################################')
T_0_n

disp('Testing endeffector frame in 0 configuration');
t1 = 0; t2 = 0; t3 = 0; t5 = 0; t6 = 0;
l1 = 4; l2 =4; l3 =4; l4 =4;l5=4; l6 =4;
subs(T_0_n)


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%     Q1 Configuration
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%substituting values into variables
t1 = pi/4; t2 = pi/6; t3 = 0; d4 =5 ; t5 = pi/3; t6 = 0;
l1 = 4; l2 =4; l3 =4; l4 =4;l5=4; l6 =4;
T_0_n1=subs(T_0_n);

disp('########################################################################')
disp('         q1 configuration is given as follows                  ')
disp('########################################################################')
disp('q1 fraction representation')
T_0_n1
disp('q1 decimal representation')
double(T_0_n1)


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%$$$$$$
%     Finding values of q2 configuration
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
%New orientation can be obtained as follows
%   -Translation about current end effector frame followed by
%   -Rotation about current z axis followed by
%   -Rotation about current y axis followed by
%   -Rotation about current z axis

% translation
trans = [1 0 0 1;
         0 1 0 2;
         0 0 1 3;
         0 0 0 1];

% Rotation
syms th1

R_z(th1) = [cos(th1) -sin(th1) 0 0;
            sin(th1) cos(th1) 0 0;
            0 0 1 0;
            0 0 0 1];

R_y(th1) = [cos(th1) 0 sin(th1) 0;
             0 1 0 0;
            -sin(th1) 0 cos(th1) 0;
             0 0 0 1];

disp('######################################################################')
disp('          q2 configuration is given as follows                        ')
disp('######################################################################')
disp('q2 fraction representation')
q2 = simplify(T_0_n1 * trans*R_z(pi/6)*R_y(pi/4)*R_z(pi/3))
disp('q2 decimal representation')
q2 = double(q2)


% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                               PROBLEM 2
% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
disp('
    #####################################################################################
    ')
disp('                              PROBLEM 2
                              ')
disp('
    ######################################################################################
    ')

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%       Finding the jacbobian
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%substituting values into variables
t1 = pi/4; t2 = pi/6; t3 = 0; d4 =5 ; t5 = pi/3; t6 = 0;
l1 = 4; l2 =4; l3 =4; l4 =4;l5=4; l6 =4;

T_0_1 = subs(T_0_1);
T_0_2 = subs(T_0_2);
T_0_3 = subs(T_0_3);
T_0_4 = subs(T_0_4);
T_0_5 = subs(T_0_5);
T_0_n = subs(T_0_n)

z0 = [0 0 1]';
z1 = T_0_1(1:3,3);
z2 = T_0_2(1:3,3);
z3 = T_0_3(1:3,3);
z4 = T_0_4(1:3,3);
z5 = T_0_5(1:3,3);
zn = T_0_n(1:3,3);

O0 = [0 0 0]';
O1 = T_0_1(1:3,4);
O2 = T_0_2(1:3,4);
O3 = T_0_3(1:3,4);
O4 = T_0_4(1:3,4);
O5 = T_0_5(1:3,4);
On = T_0_n(1:3,4);

J = [cross(z0,On-00) cross(z1,On-O1) cross(z2,On-O2) ...
     z3 cross(z4,On-O4) cross(z5,On-O5);
     z0 z1 z2 0 z4 z5;
    ];

disp('
```

```
##################################################################
')
disp('                          Jacobian                            ')
disp('##################################################################')
disp('Jacobian fraction representation')
J = simplify(J)
disp('Jacobian decimal representation')
double(J)


% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%  Equivalent representation of F(wrench) in End Effector frame
% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
T_d_e =[0 1 0 0;
        -1 0 0 -2;
         0 0 1 -1;
         0 0 0 1 ];

 T_d_t=[1 0 0 0;
        0 1 0 2;
        0 0 1 10;
        0 0 0 1 ];

 %Finding inverse of T_d_e
 R= T_d_e(1:3,1:3);
 d= T_d_e(1:3,4);
 R_T = transpose(R);

disp('##################################################################')
disp('      T_e_d(drill frame expressed in end effector frame)      ')
disp('##################################################################')
 T_e_d = [R_T -R_T*d;
          0 0 0 1]
```

```
disp('####################################################################')
disp('    T_e_t(tip frame expressed in end effector frame)      ')
disp('####################################################################')
  T_e_t =  T_e_d * T_d_t

  F =[0; 0; 10; 132.3876; 132.3876; 0]
  disp('Wrench(Force vector')
  F1 = [ F(1:3,1)] %Wrench(Force vector)
  disp('Wrench(Torque vector')
  T1 = [ F(4:6,1)] %Wrench(Torque vector)

 % Not complete

% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                               Finding joint torques
% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
disp('Wrench at end effector wrt to base frame')
Fe = [ F1 ; T1] %end effector wrench wrt to base frame
Jt = double(transpose(J)*Fe)


% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                                 PART B
% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%skew matrix
syms a b c

disp('display force')
F = [0; 0; 10; 132.3876; 132.3876;0]

S(a,b,c) =  [0 -c b;
```

```
215                 c  0 -a;
216                -b  a  0];
217
218 T_d_e =[0  1  0  0;
219         -1  0  0 -2;
220          0  0  1 -1;
221          0  0  0  1 ];
222
223  T_d_t=[1  0  0  0;
224         0  1  0  2;
225         0  0  1  10;
226         0  0  0  1 ];
227
228  disp('Finding T_n_0')
229  R= T_0_n(1:3,1:3);
230  d= T_0_n(1:3,4);
231  R_T = transpose(R);
232  T_n_0 = [R_T -R_T*d;
233           0  0  0  1];
234
235  disp('Finding T_t_d')
236  R= T_d_t(1:3,1:3);
237  d= T_d_t(1:3,4);
238  R_T = transpose(R);
239  T_t_d = [R_T -R_T*d;
240           0  0  0  1];
241
242
243  T_t_0 =  T_n_0 * T_d_e * T_t_d
244
245  R_t_0 = double(T_t_0(1:3,1:3));
246  P_t_0 = double(T_t_0(1:3,4));
247 disp('Wrench at end effector wrt to tip frame')
248 Ft = [R_t_0                                          zeros(3);
249       S(P_t_0(1),P_t_0(2),P_t_0(3))*R_t_0       R_t_0] *F;
250
251 double(Ft)
```