```
% Robot Dynamics Midterm
% Fall 2020
% Kathia Coronado


%% Problem 1
% Derive the Forward Kinematics (Ton) using the POE
approach
clear all;
clc

syms th1 th2 th3 th4 th5 th6

L=100 ;
M= [0 1 0 -3*L; 1 0 0 -L; 0 0 -1 -2*L; 0 0 0 1]; % M matrix
calculated using POE
Slist= [[0;0;0;0;0;1] [0;0;1;0;0;0;] [0;-1;0;L;0;L]
[0;0;0;-1;0;0] [1;0;0;0;L;0] [0;0;-1;L;-3*L;0]]; % Got with
POE
thetalist = [th1; th2; th3; th4; th5; th6]; % symbolic
thetas

I = eye(3);
T ={};

for i = 1 : length(thetalist)    % loop thru the joints
    w = vector_2_skew(Slist(1:3,i)); % skewmetric omega
    v = Slist(4:6,i);% v
    theta = thetalist(i); % pick that theta

    R = I + sin(theta)* w + (1-cos(theta))*w^2; % calculate
rotation matrix
    star = (I * theta+(1-cos(theta))*w+(theta-
sin(theta))*w^2)*v; % calculate big v
    T{i} = [R star; 0 0 0 1]; % construct transformation
matrix

end

fprintf('The Forward Kinematics is : \n ')
T = T{1}*T{2}*T{3}*T{4}*T{5}*T{6}*M % multiply all of the
transformations to get the HTM
pe = expand(T(1:3,4));% extract position part of the matrix
```

```
%T_home= subs(T, [th1 th2 th3 th4 th5 th6], [0 0 0 0 0 0])

%% Problem 2 : solve for position of EE at home position
T_home= subs(T, [th1 th2 th3 th4 th5 th6], [0 0 0 0 0 0])
p_home = expand(T_home(1:3,4));


fprintf('Home position of the End effector : \n ')
p_home

%% problem 3:
pb= [10; 10; 10; 1];
p0= T_home*pb;
fprintf('The vector p in the base frame is : \n ')
p0

%% problem 4 Jacobian

pe = expand(T(1:3,4));
%Jv_home=subs(Jv,[th1,th2,th3,th4,th5,th6],[0,0,0,0,0,0]);
% upper jabobian

z0= [0;0;1]
R01= [1 0 0; 0 1 0; 0 0 1];
z1= R01*z0;
R12= [cos(th2) -sin(th2) 0;sin(th2) cos(th2) 0 ; 0 0 0]*[1
0 0; 0 1 0; 0 0 1];
R02= R01*R12;
z2= R02*z0
R23= [cos(th3) -sin(th3) 0;sin(th3) cos(th3) 0 ; 0 0 0]*[0
-1 0; 0 0 -1; 1 0 0];
R03=R02*R23;
z3= R03*z0
R34=[1 0 0;0 1 0 ; 0 0 1]*[-1 0 0; 0 0 1; 0 1 0];
R04=R03*R34
z4=R04*z0
R45= [cos(th5) -sin(th5) 0;sin(th5) cos(th5) 0 ; 0 0 0]*[-1
0 0;0 1 0;0 0 -1];
R05= R04*R45
z5=R05*z0

Jv= jacobian(pe, [th1,th2,th3,th4,th5,th6]);
Jw=[z0, z1, z2, z3, z4, z5];
```

```
J= [Jv;Jw]
fprintf('Jacobian at home position : \n ');
J_home= subs(J,[th1, th2,th3,th4,th5,th6],[0,0,0,0,0,0])

%% Problem 5: find singularity
%det_J = det(J)
det_J=det(J*transpose(J))

singularity_eqn= det_J==0;
[sin_th1,sin_th2,sin_th3,sin_th4,sin_th5,sin_th6]=solve(sin
gularity_eqn, [th1,th2,th3,th4,th5,th6]);

sing_J=
subs(J,[th1,th2,th3,th4,th5,th6],[sin_th1,sin_th2,sin_th3,s
in_th4,sin_th5,sin_th6])

J_det= det(sing_J) % singularity occurs atthe home position
??


fprintf('Singularity occurs at thetas:\n')
fprintf('%f
\n',sin_th1,sin_th2,sin_th3,sin_th4,sin_th5,sin_th6)
%sin_th1,sin_th2,sin_th3,sin_th4,sin_th5,sin_th6
fprintf('At this configuration the det(J)=0 \n')

%% Problem 6: Inverse velocity. solve for the joint
velocities
pd= [10; 0 ;10];
Jv_home=subs(Jv,[th1,th2,th3,th4,th5,th6],[0,0,0,0,0,0]);
Jv_inv= pinv(Jv_home);
qd= Jv_inv*pd;
fprintf('The joint velocites are: \n')
double(qd)
fprintf('Just one solution exists \n')
%% Problem 7: Inverse position Kinematics

pd = [-350;50;-250]; % target position
q0= [1;1;1;1;1;1]; % my initial guess is the home position
qq=q0;
pe =subs(pe,[th1 th2 th3 th4 th5 th6],[qq(1) qq(2) qq(3)
qq(4) qq(5) qq(6)]);
error = pd-pe;
i = 1;
```

```
while norm(error)> 0.01
Jv_q = subs(Jv,[th1 th2 th3 th4 th5 th6],[qq(1) qq(2) qq(3)
qq(4) qq(5) qq(6)]);
delta_q = pinv(Jv_q) * error;
qq = qq +delta_q;
double(qq)
pe = subs(pe,[th1 th2 th3 th4 th5 th6],[qq(1) qq(2) qq(3)
qq(4) qq(5) qq(6)]);
error = pd - pe;
double(error)
fprintf('Iteration %f done \n', i);
i = i + 1;
end
qq


%% Extra Credit: 6x6 analytical jacobian of the arm using
xyz euler.
syms ph th ps


Rx= [1 0 0; 0 cos(ph) -sin(ph); 0 sin(ph) cos(ph)]
Ry= [cos(th) 0 sin(th); 0 1 0 ; -sin(th) 0 cos(th)]
Rz= [cos(ps) -sin(ps) 0;sin(ps) cos(ps) 0; 0 0 1]
R= Rx*Ry*Rz

wx= [1;0;0];
wy= Rx*[0;1;0];
wz= Rx*Ry*[0;0;1];
B= [wx wy wz]

ph= atan2(-T(2,3),T(3,3))
th= atan2(T(1,3),sqrt(T(1,1)^2+T(1,2)^2))
ps= atan2(-T(1,2),T(1,1))


I33 = eye(3);
zero33 = zeros(3,3);

T_alpha = [I33 zero33;
           zero33 B]
```

```
Ja = inv(T_alpha)*J


%% EXTRA CREDIT PART 2 : SOLVE FOR ANALYTICAL JACOBIAN AT
HOME

config = [0 0 0 0 0 0 ]
ph= subs(ph,[th1 th2 th3 th4 th5 th6], config)
th= subs(th,[th1 th2 th3 th4 th5 th6], config)
ps= subs(ps,[th1 th2 th3 th4 th5 th6], config)


Rx= [1 0 0; 0 cos(ph) -sin(ph); 0 sin(ph) cos(ph)]
Ry= [cos(th) 0 sin(th); 0 1 0 ; -sin(th) 0 cos(th)]
Rz= [cos(ps) -sin(ps) 0;sin(ps) cos(ps) 0; 0 0 1]
R= Rx*Ry*Rz

wx= [1;0;0];
wy= Rx*[0;1;0];
wz= Rx*Ry*[0;0;1];
B= [wx wy wz]

ph= atan2(-T(2,3),T(3,3))
th= atan2(T(1,3),sqrt(T(1,1)^2+T(1,2)^2))
ps= atan2(-T(1,2),T(1,1))


I33 = eye(3);
zero33 = zeros(3,3);

T_alpha = [I33 zero33;
           zero33 B]


Ja = inv(T_alpha)*J

% home position
J_home = subs(J, [th1 th2 th3 th4 th5 th6], config)
T_alpha_home = subs(T_alpha, [th1 th2 th3 th4 th5 th6],
config)
fprintf('The result of the Jacobian in the home position
is: \n')
Ja_home = subs(Ja,[th1 th2 th3 th4 th5 th6], config)

%% function X = vector_2_skew(x)
```

```matlab
function X = vector_2_skew(x)

X=[0 -x(3) x(2) ; x(3) 0 -x(1) ; -x(2) x(1) 0 ];

end
```