Nguyen Nguyen Anh
Id: 20205107

Homework #4

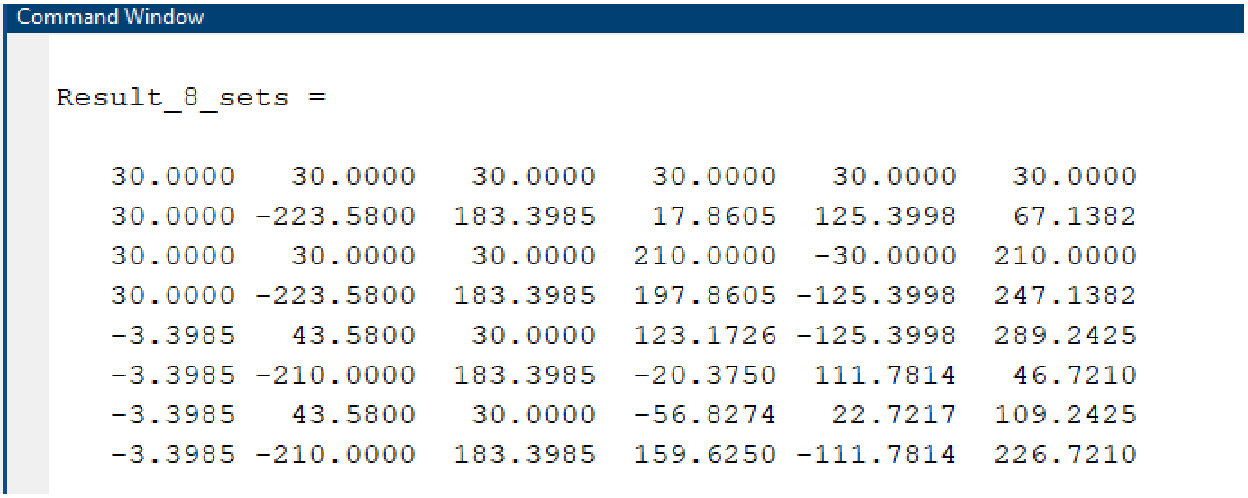
** About HW #3, because I have some mistake between [rad] and [degree], I fixed code in below:

```
clc
theta1=30*pi/180;
theta2=30*pi/180;
theta3=30*pi/180;
theta4=30*pi/180;
theta5=30*pi/180;
theta6=30*pi/180;
a2 = 1;
a3 = 0.3;
d3 = 0.5;
d4 = 1;
P6_T= [0;
       0;
       0.3;
       1];

T0_1 = [cos(theta1), -sin(theta1), 0, 0;
        sin(theta1),  cos(theta1), 0, 0;
                  0,            0, 1, 0;
                  0,            0, 0, 1];

T1_2 = [ cos(theta2), -sin(theta2), 0, 0;
                   0,            0, 1, 0;
        -sin(theta2), -cos(theta2), 0, 0;
                   0,            0, 0, 1];

T2_3 = [cos(theta3), -sin(theta3), 0, a2;
        sin(theta3),  cos(theta3), 0, 0;
                  0,            0, 1, d3;
                  0,            0, 0, 1];

T3_4 = [ cos(theta4), -sin(theta4), 0, a3;
                   0,            0, 1, d4;
        -sin(theta4), -cos(theta4), 0, 0;
                   0,            0, 0, 1];

T4_5 = [ cos(theta5), -sin(theta5),  0, 0;
                   0,            0, -1, 0;
         sin(theta5),  cos(theta5),  0, 0;
                   0,            0,  0, 1];

T5_6 = [ cos(theta6), -sin(theta6), 0, 0;
```

```
                   0,            0, 1, 0;
        -sin(theta6), -cos(theta6), 0, 0;
                   0,            0, 0, 1];
T0_6 = T0_1 * T1_2 * T2_3 * T3_4 * T4_5 * T5_6

P0_T = T0_1 * T1_2 * T2_3 * T3_4 * T4_5 * T5_6 * P6_T


%%%%%% Inverse Kinematics
r11=T0_6(1,1);
r12=T0_6(1,2);
r13=T0_6(1,3);
r21=T0_6(2,1);
r22=T0_6(2,2);
r23=T0_6(2,3);
r31=T0_6(3,1);
r32=T0_6(3,2);
r33=T0_6(3,3);

px=T0_6(1,4);
py=T0_6(2,4);
pz=T0_6(3,4);

theta1_a = atan2(py,px)-atan2(d3,(px^2+py^2-d3^2)^0.5);
theta1_b = atan2(py,px)-atan2(d3,-sqrt(px^2+py^2-d3^2));
c1_a = cos(theta1_a);
c1_b = cos(theta1_b);
s1_a = sin(theta1_a);
s1_b = sin(theta1_b);

K = (px*px+py*py+pz*pz-a2^2-a3^2-d3^2-d4^2)/(2*a2);

theta3_a = atan2(a3,d4)-atan2(K,sqrt(a3^2+d4^2-K^2));
theta3_b = atan2(a3,d4)-atan2(K,-sqrt(a3^2+d4^2-K^2));
c3_a = cos(theta3_a);
c3_b = cos(theta3_b);
s3_a = sin(theta3_a);
s3_b = sin(theta3_b);

theta23_aa = atan2((-a3-a2*c3_a)*pz-(c1_a*px+s1_a*py)*(d4-
a2*s3_a),(a2*s3_a-d4)*pz+(a3+a2*c3_a)*(c1_a*px+s1_a*py));
theta23_bb = atan2((-a3-a2*c3_b)*pz-(c1_b*px+s1_b*py)*(d4-
a2*s3_b),(a2*s3_b-d4)*pz+(a3+a2*c3_b)*(c1_b*px+s1_b*py));
theta23_ab = atan2((-a3-a2*c3_b)*pz-(c1_a*px+s1_a*py)*(d4-
a2*s3_b),(a2*s3_b-d4)*pz+(a3+a2*c3_b)*(c1_a*px+s1_a*py));
theta23_ba = atan2((-a3-a2*c3_a)*pz-(c1_b*px+s1_b*py)*(d4-
a2*s3_a),(a2*s3_a-d4)*pz+(a3+a2*c3_a)*(c1_b*px+s1_b*py));

theta2_aa = theta23_aa - theta3_a;
theta2_bb = theta23_bb - theta3_b;
theta2_ab = theta23_ab - theta3_b;
theta2_ba = theta23_ba - theta3_a;
```

```
c23_aa = cos(theta23_aa);
c23_bb = cos(theta23_bb);
c23_ab = cos(theta23_ab);
c23_ba = cos(theta23_ba);
s23_aa = sin(theta23_aa);
s23_bb = sin(theta23_bb);
s23_ab = sin(theta23_ab);
s23_ba = sin(theta23_ba);

theta4_aa1 = atan2(-r13*s1_a+r23*c1_a,-r13*c1_a*c23_aa-
r23*s1_a*c23_aa+r33*s23_aa);
theta4_bb1 = atan2(-r13*s1_b+r23*c1_b,-r13*c1_b*c23_bb-
r23*s1_b*c23_bb+r33*s23_bb);
theta4_ab1 = atan2(-r13*s1_a+r23*c1_a,-r13*c1_a*c23_ab-
r23*s1_a*c23_ab+r33*s23_ab);
theta4_ba1 = atan2(-r13*s1_b+r23*c1_b,-r13*c1_b*c23_ba-
r23*s1_b*c23_ba+r33*s23_ba);

c4_aa = cos(theta4_aa1);
c4_bb = cos(theta4_bb1);
c4_ab = cos(theta4_ab1);
c4_ba = cos(theta4_ba1);
s4_aa = sin(theta4_aa1);
s4_bb = sin(theta4_bb1);
s4_ab = sin(theta4_ab1);
s4_ba = sin(theta4_ba1);

theta4_aa2 = theta4_aa1+pi;
theta4_bb2 = theta4_bb1+pi;
theta4_ab2 = theta4_ab1+pi;
theta4_ba2 = theta4_ba1+pi;

s5_aa = -1*(r13*(c1_a*c23_aa*c4_aa+s1_a*s4_aa)+r23*(s1_a*c23_aa*c4_aa-
c1_a*s4_aa)-r33*(s23_aa*c4_aa));
s5_bb = -1*(r13*(c1_b*c23_bb*c4_bb+s1_b*s4_bb)+r23*(s1_b*c23_bb*c4_bb-
c1_b*s4_bb)-r33*(s23_bb*c4_bb));
s5_ab = -1*(r13*(c1_a*c23_ab*c4_ab+s1_a*s4_ab)+r23*(s1_a*c23_ab*c4_ab-
c1_a*s4_ab)-r33*(s23_ab*c4_ab));
s5_ba = -1*(r13*(c1_b*c23_ba*c4_ba+s1_b*s4_ba)+r23*(s1_b*c23_ba*c4_ba-
c1_b*s4_ba)-r33*(s23_ba*c4_ba));
c5_aa = r13*(-c1_a*s23_aa)+r23*(-s1_a*s23_aa)+r33*(-c23_aa);
c5_bb = r13*(-c1_b*s23_bb)+r23*(-s1_b*s23_bb)+r33*(-c23_bb);
c5_ab = r13*(-c1_a*s23_ab)+r23*(-s1_a*s23_ab)+r33*(-c23_ab);
c5_ba = r13*(-c1_b*s23_ba)+r23*(-s1_b*s23_ba)+r33*(-c23_ba);

theta5_aa1 = atan2(s5_aa,c5_aa);
theta5_bb1 = atan2(s5_bb,c5_bb);
theta5_ab1 = atan2(s5_ab,c5_ab);
theta5_ba1 = atan2(s5_ba,c5_ba);

theta5_aa2 = -theta5_aa1;
```

```
theta5_bb2 = -theta5_bb1;
theta5_ab2 = -theta5_ab1;
theta5_ba2 = -theta5_ab1;

s6_aa = -r11*(c1_a*c23_aa*s4_aa-s1_a*c4_aa) -
r21*(s1_a*c23_aa*s4_aa+c1_a*c4_aa)+r31*(s23_aa*s4_aa);
s6_bb = -r11*(c1_b*c23_bb*s4_bb-s1_b*c4_bb) -
r21*(s1_b*c23_bb*s4_bb+c1_b*c4_bb)+r31*(s23_bb*s4_bb);
s6_ab = -r11*(c1_a*c23_ab*s4_ab-s1_a*c4_ab) -
r21*(s1_a*c23_ab*s4_ab+c1_a*c4_ab)+r31*(s23_ab*s4_ab);
s6_ba = -r11*(c1_b*c23_ba*s4_ba-s1_b*c4_ba) -
r21*(s1_b*c23_ba*s4_ba+c1_b*c4_ba)+r31*(s23_ba*s4_ba);
c6_aa = r11*((c1_a*c23_aa*c4_aa+s1_a*s4_aa)*c5_aa-
c1_a*s23_aa*s5_aa)+r21*((s1_a*c23_aa*c4_aa-c1_a*s4_aa)*c5_aa-
s1_a*s23_aa*s5_aa)-r31*(s23_aa*c4_aa*c5_aa+c23_aa*s5_aa);
c6_bb = r11*((c1_b*c23_bb*c4_bb+s1_b*s4_bb)*c5_bb-
c1_b*s23_bb*s5_bb)+r21*((s1_b*c23_bb*c4_bb-c1_b*s4_bb)*c5_bb-
s1_b*s23_bb*s5_bb)-r31*(s23_bb*c4_bb*c5_bb+c23_bb*s5_bb);
c6_ab = r11*((c1_a*c23_ab*c4_ab+s1_a*s4_ab)*c5_ab-
c1_a*s23_ab*s5_ab)+r21*((s1_a*c23_ab*c4_ab-c1_a*s4_ab)*c5_ab-
s1_a*s23_ab*s5_ab)-r31*(s23_ab*c4_ab*c5_ab+c23_ab*s5_ab);
c6_ba = r11*((c1_b*c23_ba*c4_ba+s1_b*s4_ba)*c5_ba-
c1_b*s23_ba*s5_ba)+r21*((s1_b*c23_ba*c4_ba-c1_b*s4_ba)*c5_ba-
s1_b*s23_ba*s5_ba)-r31*(s23_ba*c4_ba*c5_ba+c23_ba*s5_ba);

theta6_aa1 = atan2(s6_aa,c6_aa);
theta6_bb1 = atan2(s6_bb,c6_bb);
theta6_ab1 = atan2(s6_ab,c6_ab);
theta6_ba1 = atan2(s6_ba,c6_ba);

theta6_aa2 = theta6_aa1+pi;
theta6_bb2 = theta6_bb1+pi;
theta6_ab2 = theta6_ab1+pi;
theta6_ba2 = theta6_ba1+pi;

Result_8_sets = (180/pi)*[  theta1_a,  theta2_aa,  theta3_a,
theta4_aa1,  theta5_aa1,  theta6_aa1;
                            theta1_a,  theta2_ab,  theta3_b,
theta4_ab1,  theta5_ab1,  theta6_ab1;
                            theta1_a,  theta2_aa,  theta3_a,
theta4_aa2,  theta5_aa2,  theta6_aa2;
                            theta1_a,  theta2_ab,  theta3_b,
theta4_ab2,  theta5_ab2,  theta6_ab2;
                            theta1_b,  theta2_ba,  theta3_a,
theta4_ba2,  theta5_ba2,  theta6_ba2;
                            theta1_b,  theta2_bb,  theta3_b,
theta4_bb1,  theta5_bb1,  theta6_bb1;
                            theta1_b,  theta2_ba,  theta3_a,
theta4_ba1,  theta5_ba1,  theta6_ba1;
                            theta1_b,  theta2_bb,  theta3_b,
theta4_bb2,  theta5_bb2,  theta6_bb2;]
```



```
Result_8_sets =

   30.0000   30.0000   30.0000   30.0000   30.0000   30.0000
   30.0000 -223.5800  183.3985   17.8605  125.3998   67.1382
   30.0000   30.0000   30.0000  210.0000  -30.0000  210.0000
   30.0000 -223.5800  183.3985  197.8605 -125.3998  247.1382
   -3.3985   43.5800   30.0000  123.1726 -125.3998  289.2425
   -3.3985 -210.0000  183.3985  -20.3750  111.7814   46.7210
   -3.3985   43.5800   30.0000  -56.8274   22.7217  109.2425
   -3.3985 -210.0000  183.3985  159.6250 -111.7814  226.7210
```

** HW4:

1. Linear and angular velocities of the tool can be calculated as below:

$${}^{i+1}\omega_{i+1} = {}^{i+1}_{\ \ i}R\,{}^{i}\omega_{i} + \dot{\theta}_{i+1}\,{}^{i+1}\hat{Z}_{i+1}$$

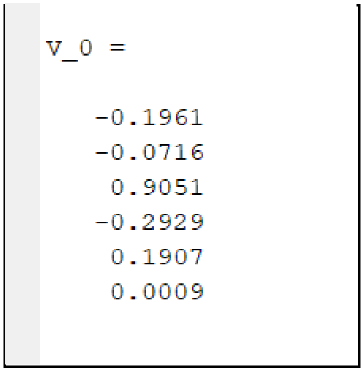
$${}^{i+1}v_{i+1} = {}^{i+1}_{\ \ i}R({}^{i}v_{i} + {}^{i}\omega_{i} \times {}^{i}P_{i+1})$$

MATLAB CODE:

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%
%%%%%%%Linear and Angular verlocities of the tool
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%
omega=[0;0;0.1]

R_0_1=[cos(theta1) -sin(theta1) 0;sin(theta1) cos(theta1) 0;0 0
1]
R_1_2=[cos(theta2) -sin(theta2) 0;0 0 1;-sin(theta2) -
cos(theta2) 0]
R_2_3=[cos(theta3) -sin(theta3) 0;sin(theta3) cos(theta3) 0;0 0
1]
R_3_4=[cos(theta4) -sin(theta4) 0;0 0 1;-sin(theta4) -
cos(theta4) 0]
R_4_5=[cos(theta5) -sin(theta5) 0;0 0 -1;sin(theta5) cos(theta5)
0]
R_5_6=[cos(theta6) -sin(theta6) 0; 0 0 1;-sin(theta6) -
cos(theta6) 0]
R_6_t=[1 0 0;0 1 0;0 0 1]

P_0_1=[0;0;0]
P_1_2=[0;0;0]
P_2_3=[a2;0;d3]
P_3_4=[a3;d4;0]
```

```
P_4_5=[ 0;0;0]
P_5_6=[0;0;0]
P_6_t=[1;1;2]
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%
% all joints verlocity
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%
w_0_0=[0;0;0]
v_0_0=[0;0;0]

w_1_1=R_0_1'*w_0_0+[0;0;0.1]
v_1_1=R_0_1'*(v_0_0+cross(w_0_0,P_0_1))

w_2_2=R_1_2'*w_1_1+[0;0;0.1]
v_2_2=R_1_2'*(v_1_1+cross(w_1_1,P_1_2))

w_3_3=R_2_3'*w_2_2+[0;0;0.1]
v_3_3=R_2_3'*(v_2_2+cross(w_2_2,P_2_3))

w_4_4=R_3_4'*w_3_3+[0;0;0.1]
v_4_4=R_3_4'*(v_3_3+cross(w_3_3,P_3_4))

w_5_5=R_4_5'*w_4_4+[0;0;0.1]
v_5_5=R_4_5'*(v_4_4+cross(w_4_4,P_4_5))

w_6_6=R_5_6'*w_5_5+[0;0;0.1]
v_6_6=R_5_6'*(v_5_5+cross(w_5_5,P_5_6))


w_t_t=R_6_t'*w_6_6+0
v_t_t=R_6_t'*(v_6_6+cross(w_6_6,P_6_t))

w_0_t=R_0_1*R_1_2*R_2_3*R_3_4*R_4_5*R_5_6*R_6_t*w_6_6
v_0_t=R_0_1*R_1_2*R_2_3*R_3_4*R_4_5*R_5_6*R_6_t*v_t_t

V_0=[v_0_t;w_0_t]
```

Result:



```
v_0 =

   -0.1961
   -0.0716
    0.9051
   -0.2929
    0.1907
    0.0009
```

2. The matrix defining the frame {i} in reference to the base frame is calculated as below:

$$ {}^{0}_{i}T = {}^{0}_{1}T\,{}^{1}_{2}T....{}^{i-1}_{i}T = \begin{bmatrix} {}^{0}_{i}R & {}^{0}_{i}P \\ 0 & 1 \end{bmatrix} \text{ where } {}^{0}_{i}R = \begin{bmatrix} {}^{0}_{i}s & {}^{0}_{i}j & {}^{0}_{i}k \end{bmatrix} $$

Then the Jacobian matrix is calculated as below:

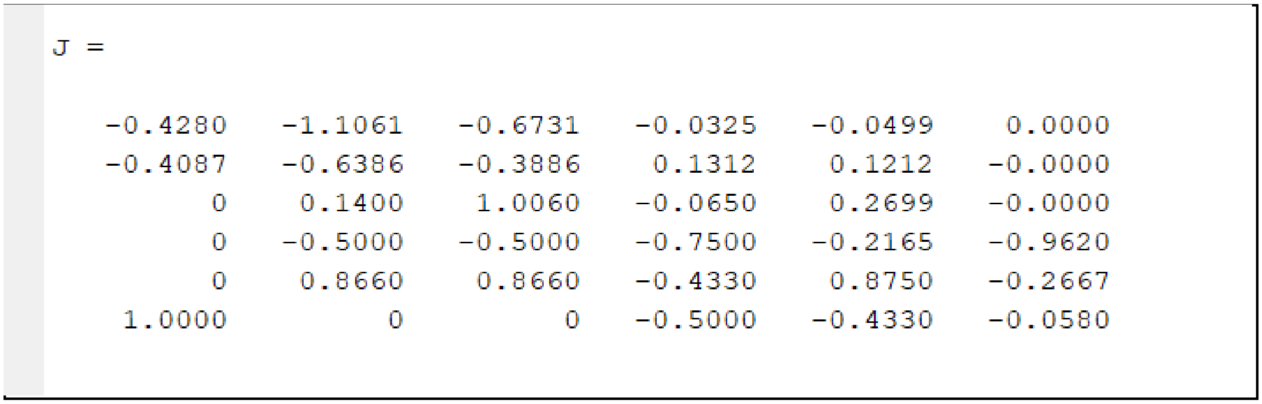
$$ J_{\theta i} = \begin{bmatrix} {}^{0}_{i}k \times ({}^{0}_{T}P - {}^{0}_{i}P) \\ {}^{0}_{i}k \end{bmatrix} $$

MATLAB CODE:

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%
%%%%%%%%%%%%Jacobian
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%
P_0_t=P0_T(1:3) ;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%
k_0_1=T0_1(1:3,3) ;
P0_1=T0_1(1:3,4) ;
J1=[cross(k_0_1,(P_0_t-P0_1));k_0_1]
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%
T0_2=T0_1*T1_2
k_0_2=T0_2(1:3,3) ;
P0_2=T0_2(1:3,4) ;
J2=[cross(k_0_2,(P_0_t-P0_2));k_0_2]
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%
T0_3=T0_1*T1_2*T2_3
k_0_3=T0_3(1:3,3) ;
P0_3=T0_3(1:3,4) ;
J3=[cross(k_0_3,(P_0_t-P0_3));k_0_3]
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%
T0_4=T0_1*T1_2*T2_3*T3_4
k_0_4=T0_4(1:3,3) ;
P0_4=T0_4(1:3,4) ;
J4=[cross(k_0_4,(P_0_t-P0_4));k_0_4]
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%
T0_5=T0_1*T1_2*T2_3*T3_4*T4_5
k_0_5=T0_5(1:3,3) ;
P0_5=T0_5(1:3,4) ;
J5=[cross(k_0_5,(P_0_t-P0_5));k_0_5]
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%
T0_6=T0_1*T1_2*T2_3*T3_4*T4_5*T5_6
k_0_6=T0_6(1:3,3) ;
P0_6=T0_6(1:3,4) ;
J6=[cross(k_0_6,(P_0_t-P0_6));k_0_6]
J=[J1 J2 J3 J4 J5 J6]
```

Result:



```
J =

   -0.4280   -1.1061   -0.6731   -0.0325   -0.0499    0.0000
   -0.4087   -0.6386   -0.3886    0.1312    0.1212   -0.0000
         0    0.1400    1.0060   -0.0650    0.2699   -0.0000
         0   -0.5000   -0.5000   -0.7500   -0.2165   -0.9620
         0    0.8660    0.8660   -0.4330    0.8750   -0.2667
    1.0000         0         0   -0.5000   -0.4330   -0.0580
```

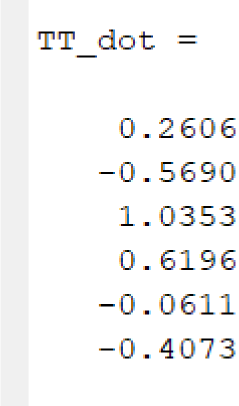
3. Inverse velocity is calculated as below:

MATLAB CODE:

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%Inverse velocity%
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

TT_dot=inv(J)*V_0
```

Result:



```
TT_dot =

    0.2606
   -0.5690
    1.0353
    0.6196
   -0.0611
   -0.4073
```