**Name: Hoang Duc Khanh Student ID: 20155889**

**Lab: Industrial Networks and Power Electronics Laboratory (INPEL)**

Class: MECHANICS AND CONTROL OF ROBOT MANIPULATORS

Instructor: Prof. Kang, Hee-Jun

Home Work 4

**Homework 3**

1. Program Forward Kinematics for Puma 560 robot with your assumed values for DRW00001f0442de and DRW00001f0442e0=> DRW00001f0442e2 then, when DRW00001f0442e4=[ 30 deg 90 90 30 30 30], Find DRW00001f0442e6
2. Program Inverse Kinematics for Puma 560 robot. Find the 8 solution sets corresponding to the DRW00001f0442e8of prob. 1. and make sure that one among your solution sets must be [ 30deg 90 90 30 30 30].

Continued from HW3, at that instance, all joint velocities are 0.1 rad / sec with the robot configuration of prob. 1. If possible, write the Matlab Program to do next problems as

1. Find the linear and angular velocities of the tool.
2. Find the Jacobian at that instant.
3. With the inverse of Jacobian and the obtained results, do velocity inverse kinematics to find the joint velocities.

Solutions

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



The result is as below:



Matlab code:

|  |
| --- |
| %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  %%%%%%%Linear and Angular velocities 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] |

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

 where 

Then the Jacobian matrix is calculated as below:



Jacobian matrix is shown as below:



Matlab code:

|  |
| --- |
| %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  %%%%%%%%%%%%Jacobian matrix calculation  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  P\_0\_t=T\_0\_t(1:3,4) ;  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  k\_0\_1=T\_0\_1(1:3,3) ;  P\_0\_1=T\_0\_1(1:3,4) ;  J1=[cross(k\_0\_1,(P\_0\_t-P\_0\_1));k\_0\_1]  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  T\_0\_2=T\_0\_1\*T\_1\_2  k\_0\_2=T\_0\_2(1:3,3) ;  P\_0\_2=T\_0\_2(1:3,4) ;  J2=[cross(k\_0\_2,(P\_0\_t-P\_0\_2));k\_0\_2]  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  T\_0\_3=T\_0\_1\*T\_1\_2\*T\_2\_3  k\_0\_3=T\_0\_3(1:3,3) ;  P\_0\_3=T\_0\_3(1:3,4) ;  J3=[cross(k\_0\_3,(P\_0\_t-P\_0\_3));k\_0\_3]  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  T\_0\_4=T\_0\_1\*T\_1\_2\*T\_2\_3\*T\_3\_4  k\_0\_4=T\_0\_4(1:3,3) ;  P\_0\_4=T\_0\_4(1:3,4) ;  J4=[cross(k\_0\_4,(P\_0\_t-P\_0\_4));k\_0\_4]  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  T\_0\_5=T\_0\_1\*T\_1\_2\*T\_2\_3\*T\_3\_4\*T\_4\_5  k\_0\_5=T\_0\_5(1:3,3) ;  P\_0\_5=T\_0\_5(1:3,4) ;  J5=[cross(k\_0\_5,(P\_0\_t-P\_0\_5));k\_0\_5]  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  T\_0\_6=T\_0\_1\*T\_1\_2\*T\_2\_3\*T\_3\_4\*T\_4\_5\*T\_5\_6  k\_0\_6=T\_0\_6(1:3,3) ;  P\_0\_6=T\_0\_6(1:3,4) ;  J6=[cross(k\_0\_6,(P\_0\_t-P\_0\_6));k\_0\_6]  J=[J1 J2 J3 J4 J5 J6] |

1. Inverse velocity is calculated as below:



Matlab code:

|  |
| --- |
| %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  %Inverse velocity%  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  TT\_dot=inv(J)\*V\_0 |