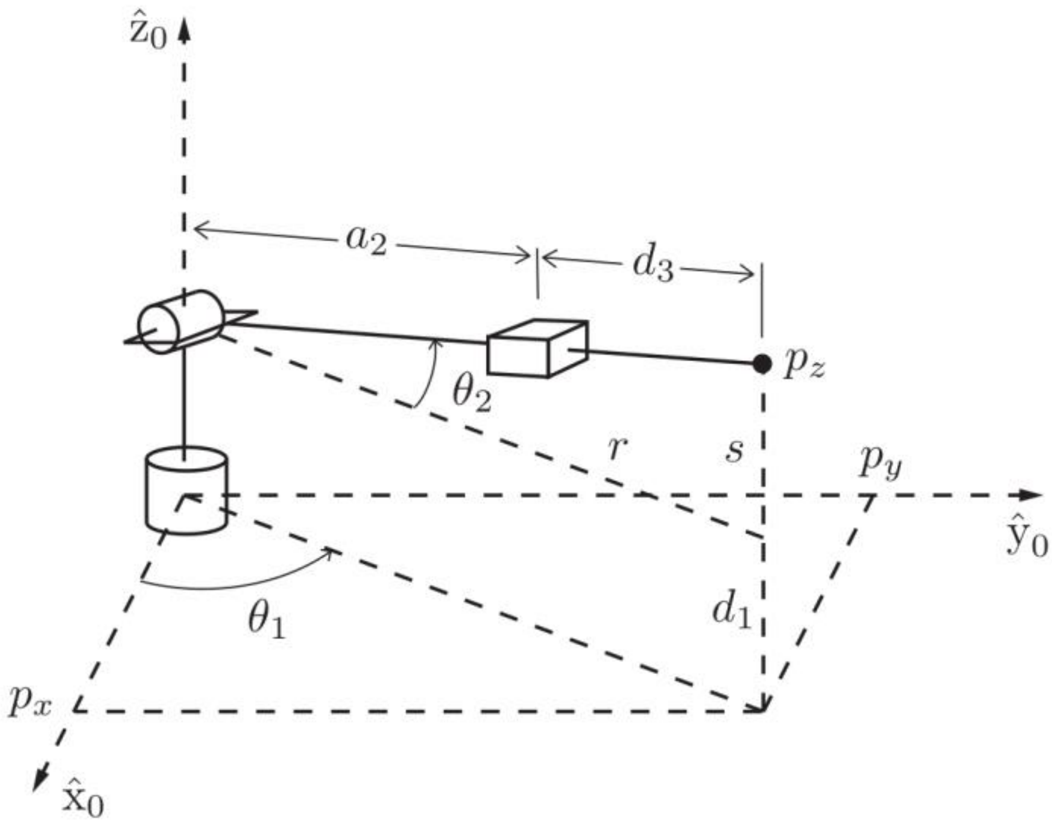
Jose Corona

Home Task 3



1. FK

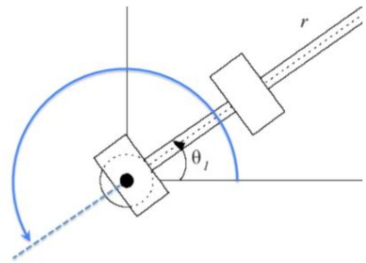
```
FK=Rz(q(1))*Tz(L(1))*Ry(-q(2))*Tx(L(2))*Tx(q(3));
```

```
[ cos(q1)*cos(q2), -sin(q1), -cos(q1)*sin(q2),    cos(q1)*cos(q2)*(a2 + q3)]
[ cos(q2)*sin(q1),  cos(q1), -sin(q1)*sin(q2),    cos(q2)*sin(q1)*(a2 + q3)]
[         sin(q2),        0,          cos(q2), d1 + a2*sin(q2) + q3*sin(q2)]
[               0,        0,                0,                            1]
```

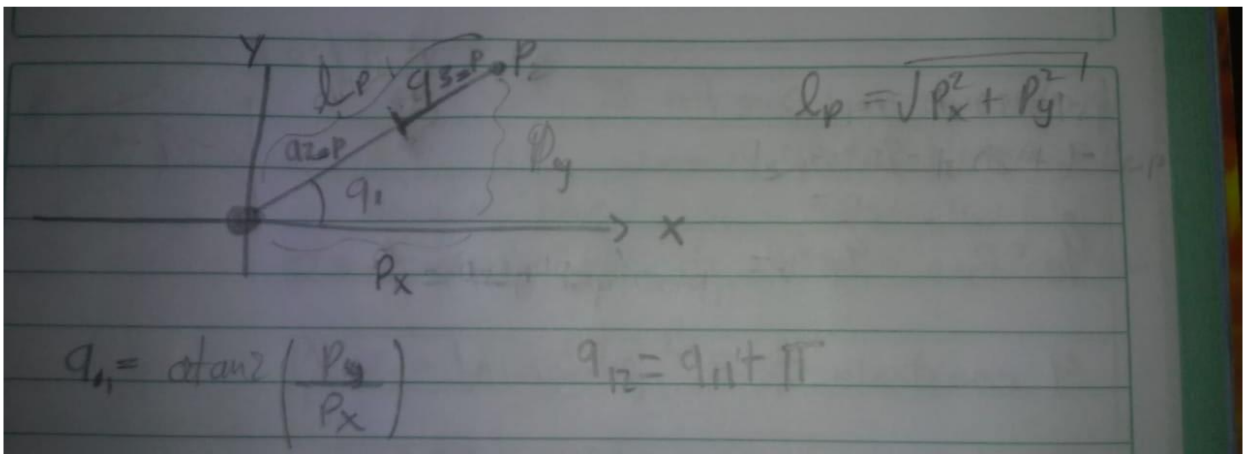
For the FK, I used “-q2” because I used the homogeneous Rotation Y. So to match the correct the coordinate system as in the diagram, is necessary to used “-q2”.

2. IK

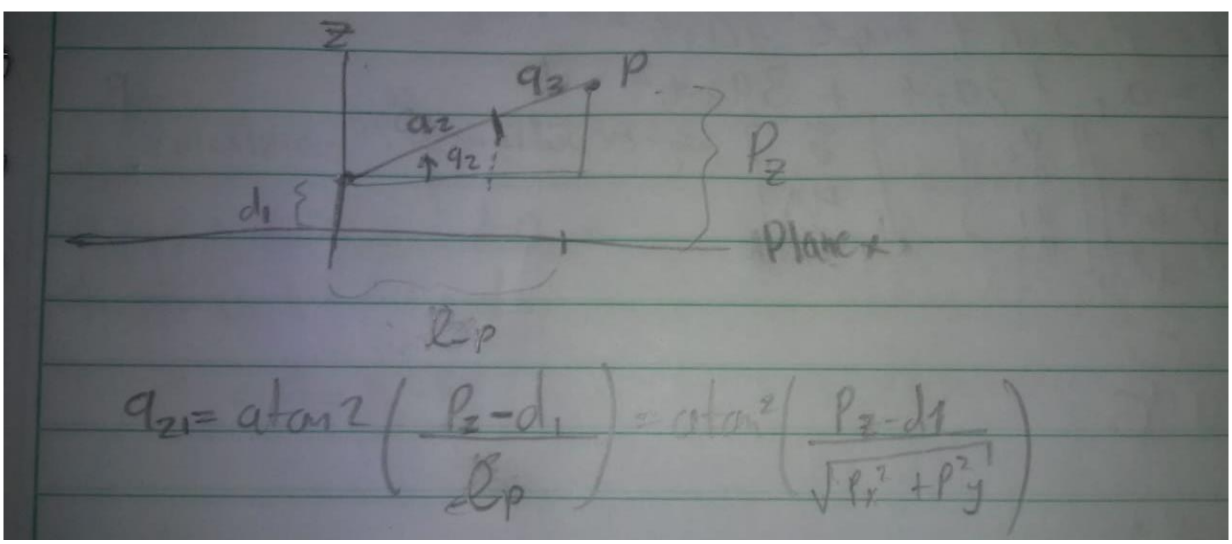
There is two solutions one for q1 and the other for PI+q1.

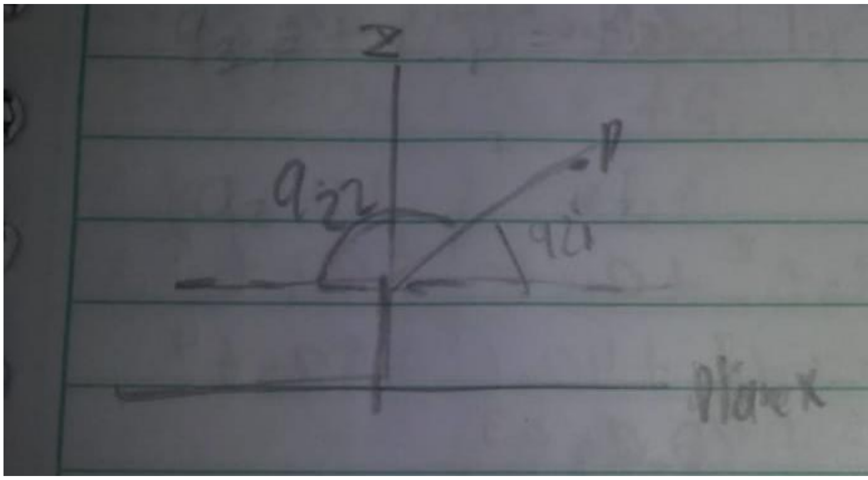


For q1:



For q2:





$$q_{22} = \pi - q_{21}$$

For q3:

$$q_3 = \sqrt{l_p^2 + (P_z - d_1)^2} - a_2$$

In MATLAB:

```
%IK
Pos=T(1:3,4) % extract position of EE
%Pos=[Px Py Pz]
%L_test=[d1 a2];
l_p=sqrt(Pos(1)^2+Pos(2)^2)

%%%%%%%%%%%%%%%%%% 1st solution
q_11=atan2(Pos(2),Pos(1));
q_3=sqrt((Pos(3)-L_test(1))^2+l_p^2)-L_test(2);
q_21=atan2((Pos(3)-L_test(1)),l_p);

q_IK_1=[q_11 q_21 q_3]

%%%%%%%%%%%%%%%% 2nd solution
q_12=pi+atan2(Pos(2),Pos(1));
q_22=pi-atan2((Pos(3)-L_test(1)),l_p);

q_IK_2=[q_12 q_22 q_3];
```

3. Jacobian

A. Derivations

```
H = Rz(q(1))*Tz(L(1))*Ry(-q(2))*Tx(L(2))*Tx(q(3));% forward kinematics

Px=H(1,4);
Py=H(2,4);
Pz=H(3,4);
```

J_1 = [J_1 contribution to linear velocity ; J_1 contribution to angular velocity]

$$\mathbf{J}_v(\mathbf{q}) = \begin{bmatrix} \frac{\partial x}{\partial q_0} & \frac{\partial x}{\partial q_1} \\ \frac{\partial y}{\partial q_0} & \frac{\partial y}{\partial q_1} \\ \frac{\partial z}{\partial q_0} & \frac{\partial z}{\partial q_1} \end{bmatrix} \qquad \mathbf{J}_\omega(\mathbf{q}) = \begin{bmatrix} \frac{\partial \omega_x}{\partial q_0} & \frac{\partial \omega_x}{\partial q_1} \\ \frac{\partial \omega_y}{\partial q_0} & \frac{\partial \omega_y}{\partial q_1} \\ \frac{\partial \omega_z}{\partial q_0} & \frac{\partial \omega_z}{\partial q_1} \end{bmatrix}$$

```
J_1 = [diff(Px,q(1)),diff(Py,q(1)),diff(Pz,q(1)),0,0,1]';
%since -q2
J_2=[-diff(Px,q(2)),-diff(Py,q(2)),-diff(Pz,q(2)),-sin(q(1)),cos(q(1)),0]';

J_3=[diff(Px,q(3)),diff(Py,q(3)),diff(Pz,q(3)),0,0,0]';

% Full Jacobian 6x3
Jq3 = [simplify(J_1), simplify(J_2), simplify(J_3)]
```

```
Jq3 =

[ -cos(q2)*sin(q1)*(a2 + q3), cos(q1)*sin(q2)*(a2 + q3), cos(q1)*cos(q2)]
[  cos(q1)*cos(q2)*(a2 + q3), sin(q1)*sin(q2)*(a2 + q3), cos(q2)*sin(q1)]
[                          0,          -cos(q2)*(a2 + q3),         sin(q2)]
[                          0,                   -sin(q1),               0]
[                          0,                    cos(q1),               0]
[                          1,                          0,               0]
```

B. Screw Matrix

$$J = \begin{bmatrix} J_1 & J_2 & \dots & J_n \end{bmatrix}$$

where if joint *(i)* is revolute

$$J_i = \begin{bmatrix} Z_{i-1} \times (O_n - O_{i-1}) \\ Z_{i-1} \end{bmatrix}$$

and if joint *(i)* is prismatic

$$J_i = \begin{bmatrix} Z_{i-1} \\ 0 \end{bmatrix}$$

```
%Finds FKs for each joint
T00= eye(4);
T01= Rz(q(1))*Tz(L(1));
T02= Rz(q(1))*Tz(L(1))*Ry(-q(2))*Tx(L(2));
T03= Rz(q(1))*Tz(L(1))*Ry(-q(2))*Tx(L(2))*Tx(q(3));

%Find Origins
O0 = T00(1:3,4);
O1 = T01(1:3,4);
O2 = T02(1:3,4);
O3 = T03(1:3,4);
```

```
%rotation (translation in case of prismatic joint) axis Z from transformation
Z0 = T00(1:3,3); % 3rd coloumn corresponds to Rz
Z1 = T01(1:3,2); % 2nd coloumn corresponds to Ry
Z2 = T02(1:3,1); % 1rd coloumn corresponds to Tx

J_1 = [cross(Z0,(O3-O0));Z0];
J_2 = [cross(Z1,(O3-O1));Z1];
J_3 = [Z2;0; 0; 0];

% Full Jacobian 6x3
Jq2 = [simplify(J_1), simplify(J_2), simplify(J_3)]
```

```
Jq2 =

[ -cos(q2)*sin(q1)*(a2 + q3), cos(q1)*sin(q2)*(a2 + q3), cos(q1)*cos(q2)]
[  cos(q1)*cos(q2)*(a2 + q3), sin(q1)*sin(q2)*(a2 + q3), cos(q2)*sin(q1)]
[                          0,          -cos(q2)*(a2 + q3),         sin(q2)]
[                          0,                   -sin(q1),               0]
[                          0,                    cos(q1),               0]
[                          1,                          0,               0]
```

C. Numerical method

For J1

```
H = Rz(q(1))*Tz(L(1))*Ry(-q(2))*Tx(L(2))*Tx(q(3));% forward kinematics
H=simplify(H);
R = simplify(H(1:3,1:3));  % extract rotation matrix
% diff by q1
Td=Rzd(q(1))*Tz(L(1))*Ry(-q(2))*Tx(L(2))*Tx(q(3))*...
    [R^-1 zeros(3,1);0 0 0 1];
J1 = [Td(1,4), Td(2,4), Td(3,4), Td(3,2), Td(1,3), Td(2,1)]' ; % extract 6 components
```

```
% diff by q2
Td=Rz(q(1))*Tz(L(1))*Ryd(-q(2))*Tx(L(2))*Tx(q(3))*...
    [R^-1 zeros(3,1);0 0 0 1];
J_2 = [Td(1,4), Td(2,4), Td(3,4), Td(3,2), Td(1,3), Td(2,1)]' ; % extract 6 components
% diff by q3
Td=Rz(q(1))*Tz(L(1))*Ry(-q(2))*Tx(L(2))*Txd(q(3))*...
    [R^-1 zeros(3,1);0 0 0 1];
J_3 = [Td(1,4), Td(2,4), Td(3,4), Td(3,2), Td(1,3), Td(2,1)]' ; % extract 6 components
```

Full Jacobian

```
% Full Jacobian 6x3
Jq1 = [simplify(J1), simplify(J2), simplify(J3)]
```

```
Jq1 =

[ -cos(q2)*sin(q1)*(a2 + q3), cos(q1)*sin(q2)*(a2 + q3), cos(q1)*cos(q2)]
[  cos(q1)*cos(q2)*(a2 + q3), sin(q1)*sin(q2)*(a2 + q3), cos(q2)*sin(q1)]
[                          0,        -cos(q2)*(a2 + q3),         sin(q2)]
[                          0,                  -sin(q1),               0]
[                          0,                   cos(q1),               0]
[                          1,                         0,               0]
```

4. Singularities:

A singularity happened when in a determinate position we lose a degree of freedom; the determinant of the Jacobian (for square matrix), is zero at singular configurations. Also, the rank decreases it number.

Using Jq2, I tried to find a singular position. I tried Gauss-Jordan elimination, but I couldn't find any singular position.

```
rref(Jq2) =

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

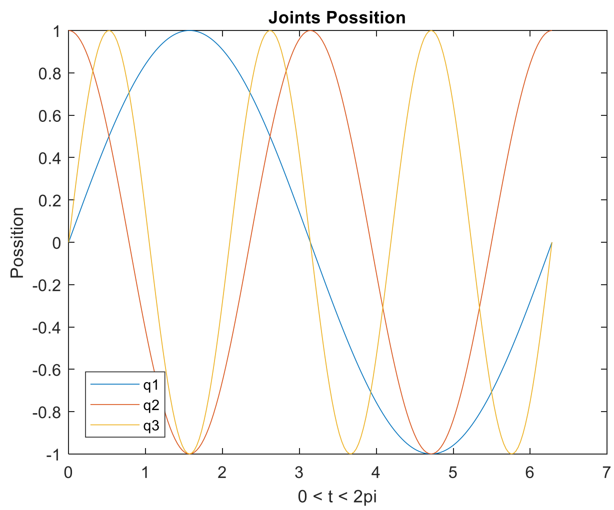
If a 3-by-3 square matrix is full rank, rank equal 3, the reduced row echelon form is an identity matrix.

If we define a specific real length for q3, we also are defining the workspace of the robot, so any point out of it, would be a singularity, since the robot could not reach it.

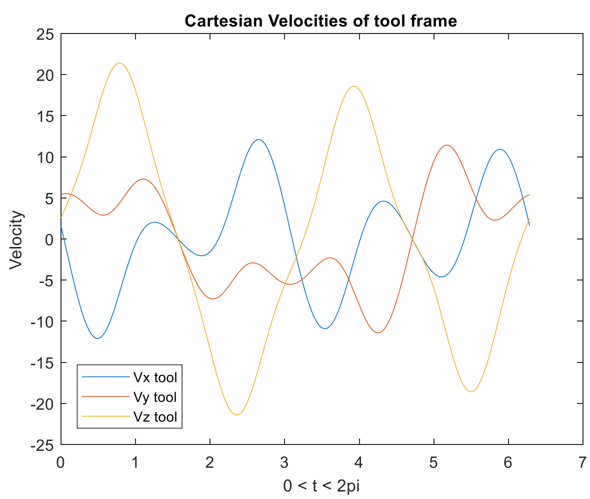
5. Velocities

$\theta_1(t) = \sin(t)$, $\theta_2(t) = \cos(2t)$, $d_3(t) = \sin(3t)$

$$\dot{\mathbf{x}} = \mathbf{J}\,\dot{\mathbf{q}},$$

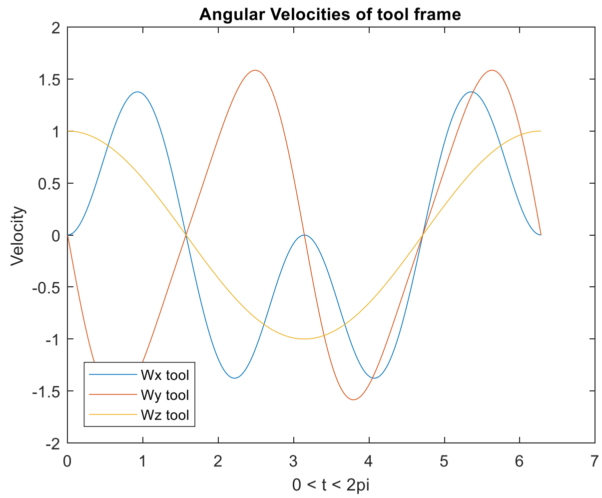


Each function has a different frequency. At the beginning the prismatic joint is inside then it goes out.



Since q2 and q3 has a bigger frequency at the beginning Px_tool is going to -x axes, with high negative magnitude velocity, and continue until reach negative max magnitude Vx_tool. The negative magnitude of Vx_tool decrease when q3, that is completely out, start to turn inside.

V_y reaches it maximum point when the robot is pointing to + y_axis, q1 is near pi/5 and q3 is almost completely out.



Git_hub:

https://github.com/Jose-R-Corona/HomeTask_3.git