

Homework 4: RRP Robot

Table of Contents

1. Forward Kinematics.....	2
2. Inverse Kinematics.....	4
3. Jacobian.....	6
Classical approach.....	6
Skew theory	8
Numerical derivatives.....	10
checking the different Jacobian methods:	12
4. Singularities.....	14
5. velocity of the tool frame.....	15

```
%Mostafa Osama Ahmed  
%Innopolis University - DoNRS course  
clear all  
close all
```

Github Link:

<https://github.com/mostafa-metwaly/DoNRs-HW4>

q1, q2, d3 are joint space variables, px, py, pz are operational space variables and Parameters

d1, a2 are known

```
% set angles and links as symbolical  
syms d1 a2 d3 q1 q2 r s Px Py Pz  
% d1=1  
% a2=1
```

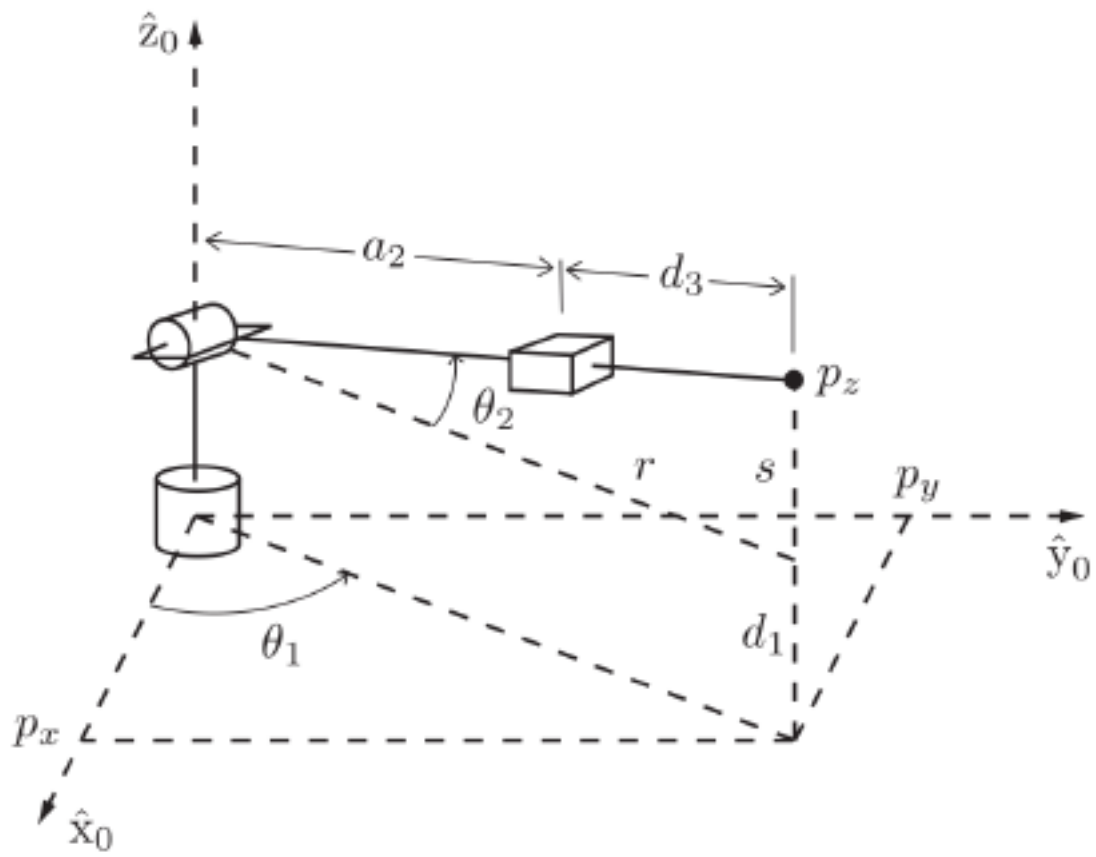
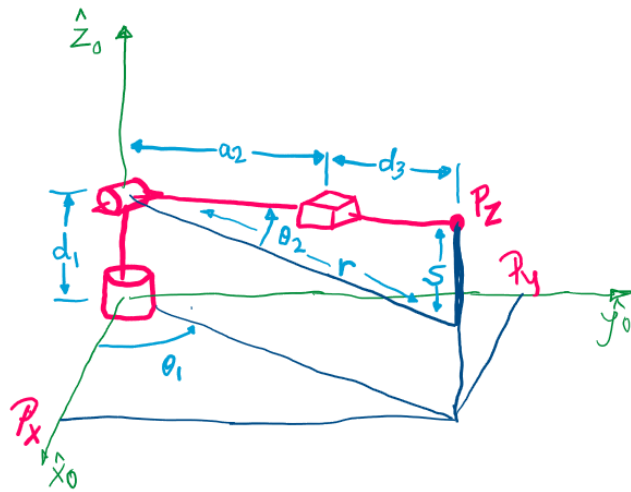


Figure 1: RRP robot.

1. Forward Kinematics

$$H = R_z(q_1) \cdot T_z(d_1) \cdot R_y(q_2) \cdot T_x(a_2) \cdot T_x(d_3)$$



```
% forward kinematics
H = simplify(Rz(q1)*Tz(d1)*Ry(-q2)*Tx(a2)*Tx(d3))
```

H =

$$\begin{pmatrix} \cos(q_1) \cos(q_2) & -\sin(q_1) & -\cos(q_1) \sin(q_2) & \cos(q_1) \cos(q_2) (a_2 + d_3) \\ \cos(q_2) \sin(q_1) & \cos(q_1) & -\sin(q_1) \sin(q_2) & \cos(q_2) \sin(q_1) (a_2 + d_3) \\ \sin(q_2) & 0 & \cos(q_2) & d_1 + a_2 \sin(q_2) + d_3 \sin(q_2) \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

```
% extract rotation matrix
R = simplify(H(1:3,1:3))
```

R =

$$\begin{pmatrix} \cos(q_1) \cos(q_2) & -\sin(q_1) & -\cos(q_1) \sin(q_2) \\ \cos(q_2) \sin(q_1) & \cos(q_1) & -\sin(q_1) \sin(q_2) \\ \sin(q_2) & 0 & \cos(q_2) \end{pmatrix}$$

Px and Py depends on the value of r :

$$Px=r*\cos(q_1) , \quad Py=r*\sin(q_1)$$

$$\text{and } r=(a_2+d_3)*\cos(q_2)$$

where for Pz value depends on s :

$$Pz=d_1+s , \quad s=(a_2+d_3)*\sin(q_2)$$

Position of the End effector for RRP robot in operational space variables.can be described by the vector T

$$\begin{bmatrix} P_x \\ P_y \\ P_z \end{bmatrix} = \begin{bmatrix} r \cos(q_1) \\ r \sin(q_1) \\ d_1 + s \end{bmatrix} = \begin{bmatrix} \cos(q_1) \cos(q_2) (a_2 + d_3) \\ \cos(q_2) \sin(q_1) (a_2 + d_3) \\ d_1 + a_2 \sin(q_2) + d_3 \sin(q_2) \end{bmatrix}$$

```
T = simplify(H(1:3,4))
```

$$T = \begin{pmatrix} \cos(q_1) \cos(q_2) (a_2 + d_3) \\ \cos(q_2) \sin(q_1) (a_2 + d_3) \\ d_1 + a_2 \sin(q_2) + d_3 \sin(q_2) \end{pmatrix}$$

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

2. Inverse Kinematics

The robot has 3 joints 2 revolute and one prismatic

Px Py Pz - sets the position of EE we have already calculated it from vector T

q1 q2 d3 - sets the orientation of EE

there are only 2 solutions for solving this q1 and another 2 solutions for q2 and d3 can be found with one way the positive value where we neglected the negative value.

since we know vector T we can calculate the angles:

q1:

q11=atan2(Px,Py) or q12= $\pi + \text{atan2}(Px,Py)$

```
disp('q1, 1st solution')
```

```
q1, 1st solution
```

```
q11=simplify(atan2(Py,Px))
```

```
q11 = angle(cos(q2) (a2 + d3) (cos(q1) + sin(q1) i))
```

```
disp('q1, 2nd solution')
```

```
q1, 2nd solution
```

```
q12=simplify(pi+atan2(Py,Px))
```

```
q12 =  $\pi + \text{angle}(\cos(q_2) (a_2 + d_3) (\cos(q_1) + \sin(q_1) i))$ 
```

we then could get the values of s, r :

$$r = \sqrt{P_x^2 + P_y^2} \quad \text{and} \quad s = P_z - d_1$$

q2:

$$q_{21} = \frac{\pi}{2} + \text{atan2}(s, r) \quad \text{or} \quad q_{22} = \frac{3\pi}{2} - \text{atan2}(s, r)$$

```
r=simplify(sqrt(Px^2+Py^2))
```

$$r = \sqrt{-(a_2 + d_3)^2 (\sin(q_2)^2 - 1)}$$

```
s=simplify(Pz-d1)
```

$$s = \sin(q_2) (a_2 + d_3)$$

```
disp('q2, for the 1st solution')
```

q2, for the 1st solution

```
q21=simplify((pi/2)+atan2(s,r))
```

q21 =

$$\frac{\pi}{2} + \text{atan2}(\sin(q_2) (a_2 + d_3), \sqrt{-(a_2 + d_3)^2 (\sin(q_2)^2 - 1)})$$

```
disp('q2, for the 2nd solution')
```

q2, for the 2nd solution

```
q22=simplify((3*pi/2)-atan2(s,r))
```

q22 =

$$\frac{3\pi}{2} - \text{atan2}(\sin(q_2) (a_2 + d_3), \sqrt{-(a_2 + d_3)^2 (\sin(q_2)^2 - 1)})$$

d3 can be found by:

$$d_3 = \sqrt{r^2 + s^2} - a_2$$

```
disp('q3, same for both solutions')
```

q3, same for both solutions

```
d3=simplify(sqrt(r^2+s^2)-a2)
```

$$d_3 = \sqrt{(a_2 + d_3)^2} - a_2$$

Orientation

$$q = \begin{bmatrix} q_1 \\ q_2 \\ d_3 \end{bmatrix} = \begin{bmatrix} \text{atan2}(P_y, P_x) \\ \frac{\pi}{2} + \text{atan2}(s, r) \\ \sqrt{r^2 + s^2} - a_2 \end{bmatrix} \quad \text{or} \quad q = \begin{bmatrix} q_1 \\ q_2 \\ d_3 \end{bmatrix} = \begin{bmatrix} \pi + \text{atan2}(P_y, P_x) \\ \frac{3\pi}{2} - \text{atan2}(s, r) \\ \sqrt{r^2 + s^2} - a_2 \end{bmatrix}$$

$$q_1 = [q_{11} \ q_{12} \ d_3]$$

$$q_1 = \left(\text{angle}(\cos(q_2) (a_2 + d_3) (\cos(q_1) + \sin(q_1) i)) \quad \pi + \text{angle}(\cos(q_2) (a_2 + d_3) (\cos(q_1) + \sin(q_1) i)) \quad \sqrt{a_2} \right)$$

$$q_2 = [q_{21} \ q_{22} \ d_3]$$

$$q_2 =$$

$$\left(\frac{\pi}{2} + \text{atan2}(\sin(q_2) (a_2 + d_3), \sqrt{-(a_2 + d_3)^2 (\sin(q_2)^2 - 1)}) \quad \frac{3\pi}{2} - \text{atan2}(\sin(q_2) (a_2 + d_3), \sqrt{-(a_2 + d_3)^2 (\sin(q_2)^2 - 1)}) \right)$$

$$q = [q_1 \ ; \ q_2]$$

$$q =$$

$$\begin{pmatrix} q_1 \\ q_2 \end{pmatrix}$$

syms q1 q2 d3 a2 d1

checking1=simplify(subs(q_1,{d1,a2,q1,q2,d3},{1,1,pi,pi,2}))

$$\text{checking1} = (0 \quad \pi \quad 2)$$

checking2=simplify(subs(q_2,{d1,a2,q1,q2,d3},{1,1,pi,pi,2}))

$$\text{checking2} =$$

$$\left(\frac{\pi}{2} \quad \frac{3\pi}{2} \quad 2 \right)$$

3. Jacobian

Classical approach

the 6-vector consisting of the linear and angular velocities of the end-effector

$$\begin{bmatrix} \dot{P}_x \\ \dot{P}_y \\ \dot{P}_z \\ \dot{\omega}_X \\ \dot{\omega}_X \\ \dot{\omega}_X \end{bmatrix}$$

Jacobian equal the partial derivative of vector T wrt each of the q1,q2,d3

$$J = \begin{bmatrix} J_v \\ J_\omega \end{bmatrix}$$

we have calculated the linear Jacobian part J_v

$$J_v = \begin{bmatrix} \delta P_x / \delta q_1 & \delta P_x / \delta q_2 & \delta P_x / \delta d_3 \\ \delta P_y / \delta q_1 & \delta P_y / \delta q_2 & \delta P_y / \delta d_3 \\ \delta P_z / \delta q_1 & \delta P_z / \delta q_2 & \delta P_z / \delta d_3 \end{bmatrix}$$

```
syms d1 a2 d3 q1 q2 r s Px Py Pz real
```

```
J1=diff(T,q1)
```

$$J1 = \begin{pmatrix} -\cos(q_2) \sin(q_1) (a_2 + d_3) \\ \cos(q_1) \cos(q_2) (a_2 + d_3) \\ 0 \end{pmatrix}$$

```
J2=diff(T,q2)
```

$$J2 = \begin{pmatrix} -\cos(q_1) \sin(q_2) (a_2 + d_3) \\ -\sin(q_1) \sin(q_2) (a_2 + d_3) \\ a_2 \cos(q_2) + d_3 \cos(q_2) \end{pmatrix}$$

```
J3=diff(T,d3)
```

$$J3 = \begin{pmatrix} \cos(q_1) \cos(q_2) \\ \cos(q_2) \sin(q_1) \\ \sin(q_2) \end{pmatrix}$$

$$J_w = [\theta, -\sin(q_1), \theta; \theta, \cos(q_1), \theta;]$$

```
1, 0, 0];
Jq1 = [simplify(J1), simplify(-J2), simplify(J3) ; Jw]
```

Jq1 =

$$\begin{pmatrix} -\cos(q_2) \sin(q_1) (a_2 + d_3) & \cos(q_1) \sin(q_2) (a_2 + d_3) & \cos(q_1) \cos(q_2) \\ \cos(q_1) \cos(q_2) (a_2 + d_3) & \sin(q_1) \sin(q_2) (a_2 + d_3) & \cos(q_2) \sin(q_1) \\ 0 & -\cos(q_2) (a_2 + d_3) & \sin(q_2) \\ 0 & -\sin(q_1) & 0 \\ 0 & \cos(q_1) & 0 \\ 1 & 0 & 0 \end{pmatrix}$$

Skew theory

or geometric approach

First find origin of each joint

FK:

```
T00= eye(4)
```

T00 = 4x4

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

```
T01= Rz(q1)*Tz(d1)
```

T01 =

$$\begin{pmatrix} \cos(q_1) & -\sin(q_1) & 0 & 0 \\ \sin(q_1) & \cos(q_1) & 0 & 0 \\ 0 & 0 & 1 & d_1 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

```
T02= Rz(q1)*Tz(d1)*Ry(-q2)
```

T02 =

$$\begin{pmatrix} \cos(q_1) \cos(q_2) & -\sin(q_1) & -\cos(q_1) \sin(q_2) & 0 \\ \cos(q_2) \sin(q_1) & \cos(q_1) & -\sin(q_1) \sin(q_2) & 0 \\ \sin(q_2) & 0 & \cos(q_2) & d_1 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

```
T03= Rz(q1)*Tz(d1)*Ry(-q2)*Tx(a2)*Tx(d3)
```

T03 =

$$\begin{pmatrix} \cos(q_1) \cos(q_2) & -\sin(q_1) & -\cos(q_1) \sin(q_2) & a_2 \cos(q_1) \cos(q_2) + d_3 \cos(q_1) \cos(q_2) \\ \cos(q_2) \sin(q_1) & \cos(q_1) & -\sin(q_1) \sin(q_2) & a_2 \cos(q_2) \sin(q_1) + d_3 \cos(q_2) \sin(q_1) \\ \sin(q_2) & 0 & \cos(q_2) & d_1 + a_2 \sin(q_2) + d_3 \sin(q_2) \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Origins as position component of FK

$$O0 = T00(1:3,4)$$

$$O0 = \begin{matrix} 3 \times 1 \\ 0 \\ 0 \\ 0 \end{matrix}$$

$$O1 = T01(1:3,4)$$

$$O1 = \begin{pmatrix} 0 \\ 0 \\ d_1 \end{pmatrix}$$

$$O2 = T02(1:3,4)$$

$$O2 = \begin{pmatrix} 0 \\ 0 \\ d_1 \end{pmatrix}$$

$$O3 = T03(1:3,4)$$

$$O3 = \begin{pmatrix} a_2 \cos(q_1) \cos(q_2) + d_3 \cos(q_1) \cos(q_2) \\ a_2 \cos(q_2) \sin(q_1) + d_3 \cos(q_2) \sin(q_1) \\ d_1 + a_2 \sin(q_2) + d_3 \sin(q_2) \end{pmatrix}$$

Find rotation (translation in case of prismatic joint) axis Z from transformation, note the column! Its should correspond to the joint axis

$$Z0 = T00(1:3,3) \text{ \% 3rd column corresponds to Rz}$$

$$Z0 = \begin{matrix} 3 \times 1 \\ 0 \\ 0 \\ 1 \end{matrix}$$

$$Z1 = T01(1:3,2) \text{ \% 2nd column corresponds to Ry}$$

$$Z1 =$$

$$\begin{pmatrix} -\sin(q_1) \\ \cos(q_1) \\ 0 \end{pmatrix}$$

```
Z2 = T02(1:3,1) % 1st coloumn corresponds to Tx
```

Z2 =

$$\begin{pmatrix} \cos(q_1) \cos(q_2) \\ \cos(q_2) \sin(q_1) \\ \sin(q_2) \end{pmatrix}$$

Full Jacobian

for revolute joint:

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

for prismatic joint:

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

```
J1 = [cross(Z0,(O3-O0));Z0];
J2 = [cross(Z1,(O3-O1));Z1];
J3 = [Z2;0; 0; 0];
```

```
Jq2 = [simplify(J1), simplify(J2), simplify(J3)]
```

Jq2 =

$$\begin{pmatrix} -\cos(q_2) \sin(q_1) (a_2 + d_3) & \cos(q_1) \sin(q_2) (a_2 + d_3) & \cos(q_1) \cos(q_2) \\ \cos(q_1) \cos(q_2) (a_2 + d_3) & \sin(q_1) \sin(q_2) (a_2 + d_3) & \cos(q_2) \sin(q_1) \\ 0 & -\cos(q_2) (a_2 + d_3) & \sin(q_2) \\ 0 & -\sin(q_1) & 0 \\ 0 & \cos(q_1) & 0 \\ 1 & 0 & 0 \end{pmatrix}$$

Numerical derivatives

```
%Same could be done for other Jacobian columns
```

```

% forward kinematics
H = Rz(q1)*Tz(d1)*Ry(-q2)*Tx(a2)*Tx(d3);
H=simplify(H);
% extract rotation matrix
R = simplify(H(1:3,1:3));
% diff by q1
Td= Rzd(q1)*Tz(d1)*Ry(-q2)*Tx(a2)*Tx(d3)*...
    [R^-1 zeros(3,1);0 0 0 1];
% extract 6 components from 4x4 Td matrix to Jacobian 1st column
J1 = [Td(1,4), Td(2,4), Td(3,4), Td(3,2), Td(1,3), Td(2,1)]'

```

J1 =

$$\begin{pmatrix} -a_2 \cos(q_2) \sin(q_1) - d_3 \cos(q_2) \sin(q_1) \\ a_2 \cos(q_1) \cos(q_2) + d_3 \cos(q_1) \cos(q_2) \\ 0 \\ 0 \\ 0 \\ \frac{\sin(q_1)^2}{\cos(q_1)^2 + \sin(q_1)^2} + \frac{\sigma_3}{\sigma_1} + \frac{\sigma_2}{\sigma_1} \end{pmatrix}$$

where

$$\sigma_1 = \sigma_3 + \sigma_2 + \cos(q_2)^2 \sin(q_1)^2 + \sin(q_1)^2 \sin(q_2)^2$$

$$\sigma_2 = \cos(q_1)^2 \sin(q_2)^2$$

$$\sigma_3 = \cos(q_1)^2 \cos(q_2)^2$$

```

% diff by q2
Td= Rz(q1)*Tz(d1)*Ryd(-q2)*Tx(a2)*Tx(d3)*...
    [R^-1 zeros(3,1);0 0 0 1];
% extract 6 components from 4x4 Td matrix to Jacobian 2nd column
J2 = [Td(1,4), Td(2,4), Td(3,4), Td(3,2), Td(1,3), Td(2,1)]'

```

J2 =

$$\begin{pmatrix} a_2 \cos(q_1) \sin(q_2) + d_3 \cos(q_1) \sin(q_2) \\ a_2 \sin(q_1) \sin(q_2) + d_3 \sin(q_1) \sin(q_2) \\ -a_2 \cos(q_2) - d_3 \cos(q_2) \\ -\frac{\cos(q_2)^2 \sin(q_1)}{\sigma_1} - \frac{\sin(q_1) \sin(q_2)^2}{\sigma_1} \\ \frac{\cos(q_1) \cos(q_2)^2}{\sigma_2} + \frac{\cos(q_1) \sin(q_2)^2}{\sigma_2} \\ 0 \end{pmatrix}$$

where

$$\sigma_1 = \cos(q_1)^2 \cos(q_2)^2 + \cos(q_1)^2 \sin(q_2)^2 + \cos(q_2)^2 \sin(q_1)^2 + \sin(q_1)^2 \sin(q_2)^2$$

$$\sigma_2 = \cos(q_2)^2 + \sin(q_2)^2$$

```
% diff by q3
Td= Rz(q1)*Tz(d1)*Ry(-q2)*Tx(a2)*Txd(d3)*...
    [R^-1 zeros(3,1);0 0 0 1];
% extract 6 components from 4x4 Td matrix to Jacobian 3rd column
J3 = [Td(1,4), Td(2,4), Td(3,4), Td(3,2), Td(1,3), Td(2,1)]'
```

J3 =

$$\begin{pmatrix} \cos(q_1) \cos(q_2) \\ \cos(q_2) \sin(q_1) \\ \sin(q_2) \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

```
% Full Jacobian 6x6
Jq3 = [simplify(J1), simplify(J2), simplify(J3)]
```

Jq3 =

$$\begin{pmatrix} -\cos(q_2) \sin(q_1) (a_2 + d_3) & \cos(q_1) \sin(q_2) (a_2 + d_3) & \cos(q_1) \cos(q_2) \\ \cos(q_1) \cos(q_2) (a_2 + d_3) & \sin(q_1) \sin(q_2) (a_2 + d_3) & \cos(q_2) \sin(q_1) \\ 0 & -\cos(q_2) (a_2 + d_3) & \sin(q_2) \\ 0 & -\sin(q_1) & 0 \\ 0 & \cos(q_1) & 0 \\ 1 & 0 & 0 \end{pmatrix}$$

checking the different Jacobian methods:

```
check1=simplify(Jq1-Jq2)
```

check1 =

$$\begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

```
check2=simplify(Jq1-Jq3)
```

check2 =

$$\begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

```
check3=simplify(Jq2-Jq1)
```

check3 =

$$\begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

```
check4=simplify(Jq2-Jq3)
```

check4 =

$$\begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

```
check5=simplify(Jq3-Jq1)
```

check5 =

$$\begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

```
check6=simplify(Jq3-Jq2)
```

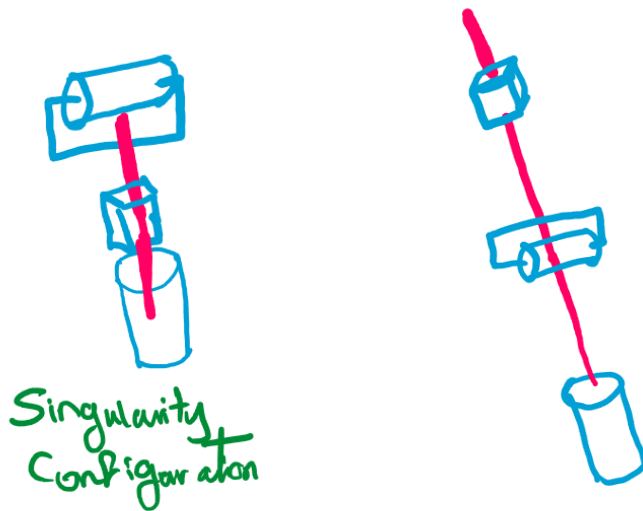
check6 =

$$\begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

then all the Jacobian Methods are the same.

4. Singularities

for Jacobian singularity analysis we need to get the determinant of the jacobian matrix in order to estimate the case which will make the matrix equal zero and we will calculate it for the upper part which is the linear part



```
analyze1=simplify(det(Jq3(1:3,1:3)))
```

$$\text{analyze1} = -\cos(q_2) (a_2 + d_3)^2$$

$$\det Jq3_v = \cos(q_2) (a_2 + d_3)^2$$

this equation from the det of the matrix will be equal to zero in two cases only when

when case1: $d_3 = -a_2$ and case2: $\cos(q_2) = 0$

case1:

d_3 can be found by:

$d_3 = \sqrt{r^2 + s^2} - a_2$ so this equation will be equal to zero if the square root part is equal to zero that means r and s are equal to zero where

$$r = \sqrt{P_x^2 + P_y^2} \quad \text{and} \quad s = P_z - d_1$$

this could happen if P_x and P_y points are equal to zero which is the origin point of the robot.

case2:

when $\cos(q_2)=0$ which happens when q_2 is equal to $\pi/2$ or $-\pi/2$ and this will reduce the number of DOF by one and the endeffector position is intersecting the Z axis and any rotation around the robot base will not affect the position of end effector.

5. velocity of the tool frame

joint variables are changing with time as follows:

$$q_1(t) = \sin(t); \quad q_2(t) = \cos(2t); \quad d_3(t) = \sin(3t)$$

we will substitute in the jacobian and multiply them by the derivatives of these equations

```
syms q1_t q2_t d3_t t real
q1_t = sin(t); q2_t = cos(2*t); d3_t = sin(3*t);

J_t=simplify(subs(Jq3,{d1,a2,q1,q2,d3},{1,1,q1_t,q2_t,d3_t}));
q=[q1_t q2_t d3_t]';
q_dot=diff(q)
```

q_dot =

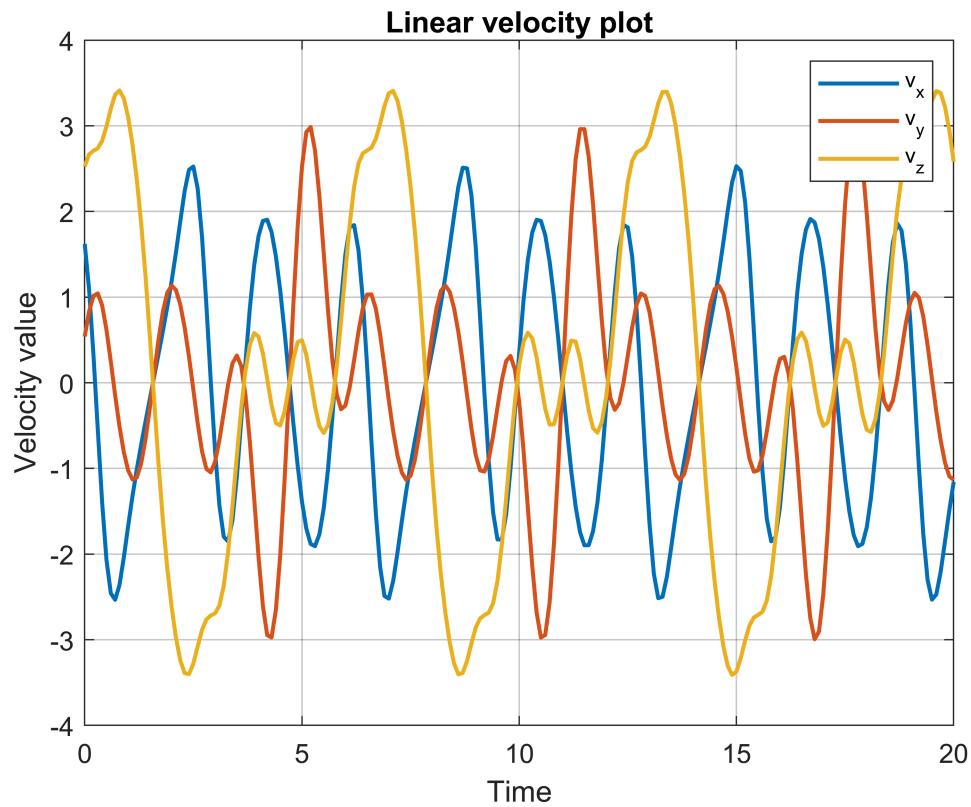
$$\begin{pmatrix} \cos(t) \\ -2 \sin(2t) \\ 3 \cos(3t) \end{pmatrix}$$

getting there derivative:

$$q_1(t) = \cos(t); \quad q_2(t) = -2 \sin(2t); \quad d_3(t) = 3 \cos(3t)$$

```
Ai= simplify(J_t*q_dot);
time=0:0.1:20;

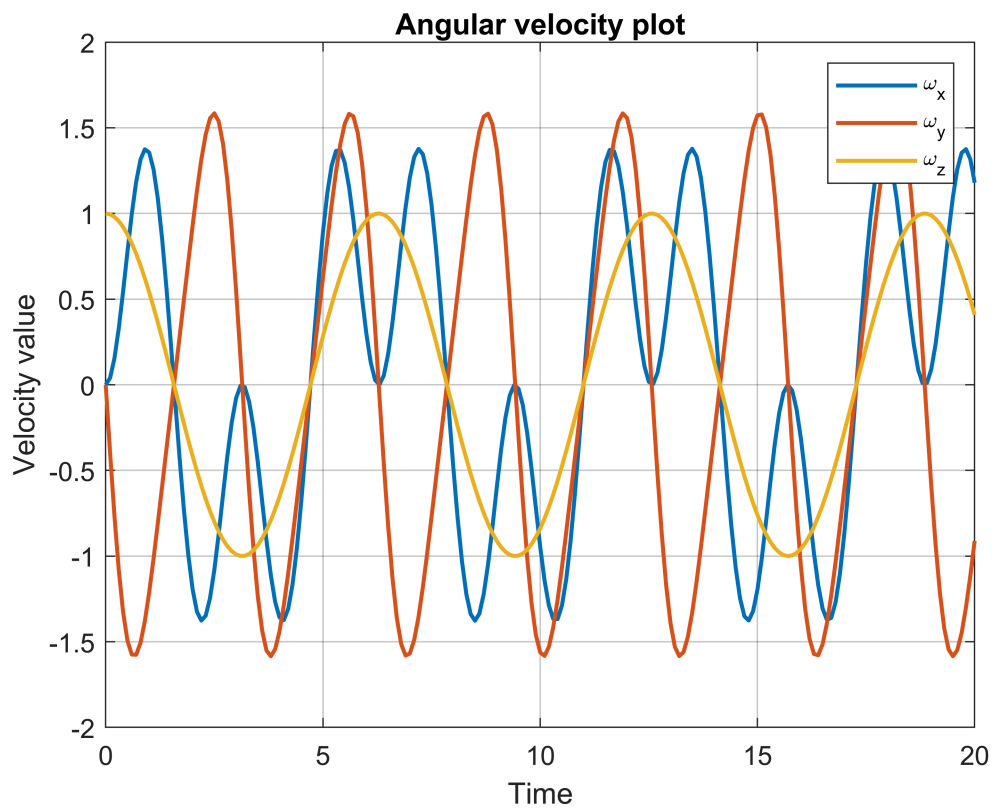
Ai_t=subs(Ai,{t},{time});
fig=figure(1);
set(gcf,'color','w');
plot(time, Ai_t(1:3, :), 'LineWidth', 1.5)
grid on
xlabel('Time')
ylabel('Velocity value')
title('Linear velocity plot')
legend('v_x', 'v_y', 'v_z', 'FontSize', 8)
frame = getframe(fig);
```



```
im = frame2im(frame);
[img,map] = rgb2ind(im,256);
imwrite(img,map,'linear_velocities.png','png');
```

Linear velocity graph is shown

```
fig = figure(2);
set(gcf,'color','w');
plot(time, Ai_t(4:6, :), 'LineWidth', 1.5)
grid on
xlabel('Time')
ylabel('Velocity value')
title('Angular velocity plot')
legend('\omega_x', '\omega_y', '\omega_z', 'FontSize', 8)
frame = getframe(fig);
```

```
im = frame2im(frame);  
[img,map] = rgb2ind(im,256);  
imwrite(img,map,'angular_velocities.png','png');
```