Dynamics of Nonlinear Robotics Systems

Home Assignment 3

Submitted by: **Mohamed Moustafa Elsayed Ahmed**

Email: **o.ahmed@innopolis.university**

Program: **Robotics**

Date:

9/10/2020

1- Forward Kinematics

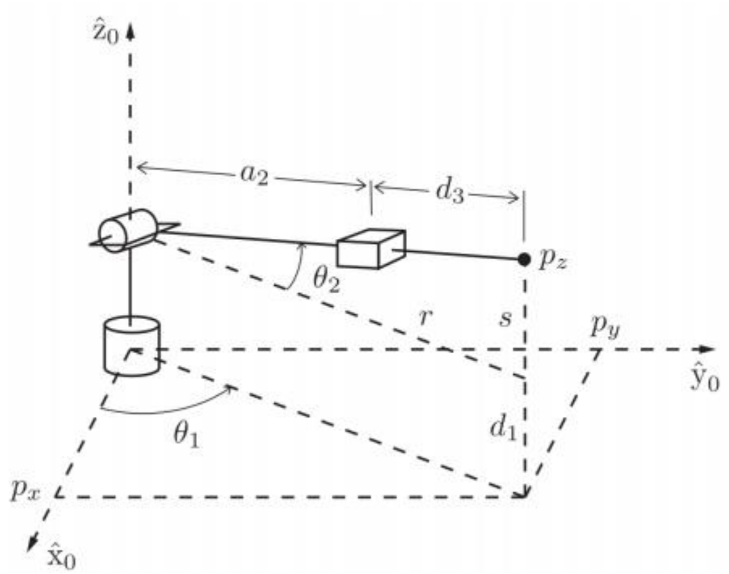


Figure 1: RRP robot

The Px and Py coordinate depend on l value:

$$Px = l\ \cos\theta_1 \quad (1)$$

$$Py = l\ \sin\theta_1 \quad (2)$$

$$l = (a2 + d3)\ \cos\theta_2 \quad (3)$$

For Pz :

$$Pz = d1 + s \quad (4)$$

$$s = (a2 + d3)\ \sin\theta_2 \quad (5)$$

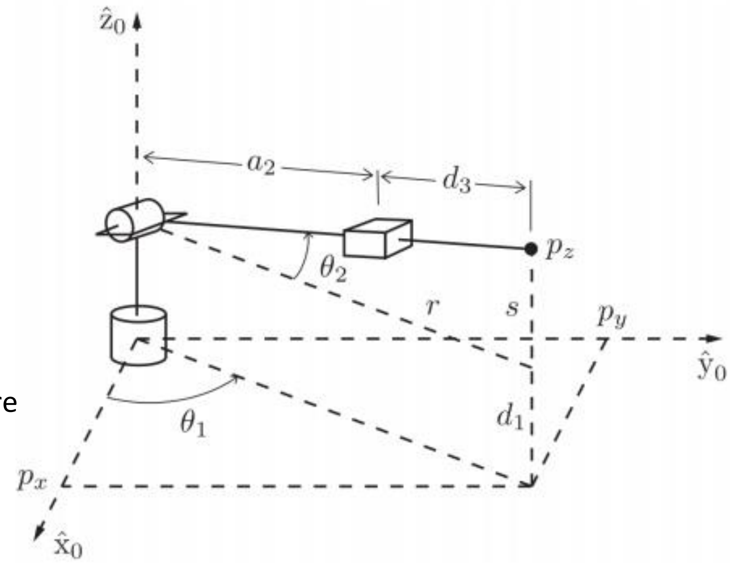
Thus position of the end-effector for RRP robot can be described by the vector

$$\begin{bmatrix} Px \\ Py \\ Pz \end{bmatrix} = \begin{bmatrix} l\ \cos\theta \\ l\ \sin\theta \\ d1 + s \end{bmatrix} = \begin{bmatrix} l(a2 + d3)\cos\theta_2\cos\theta_1 \\ l(a2 + d3)\cos\theta_2\sin\theta_1 \\ d1 + (a2 + d3)\ \sin\theta_2 \end{bmatrix}$$

Forward Kinematics:

$$H = Rz(q1) * Tz(d1) * Ry(q2) * Tx(a2) * Tx(d3);$$

2- Inverse Kinematics :



Our robot has 4 Solutions:

Imagine the given position in the following figure

$$q_1 = \text{atan2}\frac{Py}{Px}$$

$$q_2 = \text{atan2}\frac{(d1 - Pz)}{(P_x^2 + P_y^2)}$$

$$d_3 = \sqrt{(P_x^2 + P_y^2) + (d1 - Pz)^2} \text{ - a2}$$

1st Solution : $\theta_1 = q_1 ; \theta_2 = q_2 ; d3 = d3;$

2nd Solution : $\theta_{1\_2} = \theta_1 + pi ; \theta_{2\_2} = pi - \theta_2 ; d3 = d3;$

3rd Solution : $\theta_{1\_3} = \theta_1 ; \theta_{2\_3} = pi + \theta_2 ; d3 = -(d3 + 2 * a2);$

4th Solution : $\theta_1 = \theta_1 + pi ; \theta_2 = -\theta_2 ; d3 = -(d3 + 2 * a2);$

3- Comparing Jacobin results for all approaches:

a) Numerical Method:

```
J_Numerical =

[ -cos(q2)*sin(q1)*(d3 + 1), -cos(q1)*sin(q2)*(d3 + 1), cos(q1)*cos(q2)]
[  cos(q1)*cos(q2)*(d3 + 1), -sin(q1)*sin(q2)*(d3 + 1), cos(q2)*sin(q1)]
[                         0,          -cos(q2)*(d3 + 1),         -sin(q2)]
[                         0,                  -sin(q1),               0]
[                         0,                   cos(q1),               0]
[                         1,                         0,               0]
```

b) Skew Method:

```
J_Skew =

[ -cos(q2)*sin(q1)*(d3 + 1), -cos(q1)*sin(q2)*(d3 + 1), cos(q1)*cos(q2)]
[  cos(q1)*cos(q2)*(d3 + 1), -sin(q1)*sin(q2)*(d3 + 1), cos(q2)*sin(q1)]
[                         0,          -cos(q2)*(d3 + 1),         -sin(q2)]
[                         0,                  -sin(q1),               0]
[                         0,                   cos(q1),               0]
[                         1,                         0,               0]
```

c) Partial Derivative Method:

```
J_PD =

[ -cos(q2)*sin(q1)*(d3 + 1), -cos(q1)*sin(q2)*(d3 + 1), cos(q1)*cos(q2)]
[  cos(q1)*cos(q2)*(d3 + 1), -sin(q1)*sin(q2)*(d3 + 1), cos(q2)*sin(q1)]
[                         0,     - cos(q2) - d3*cos(q2),         -sin(q2)]
[                         0,                  -sin(q1),               0]
[                         0,                   cos(q1),               0]
[                         1,                         0,               0]
```

d) Comparing Results:

```
>> J_Skew-J_Numerical

ans =

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

```
>> simplify(J_Skew-J_PD)

ans =

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

4- Analyze the Jacobin for singularities:

To obtain singularity from Jacobin matrix we need to obtain determinant of it

So we will take first 3 rows and first 3 columns , get there determinant in symbolic form

J= J_PD(1:3,1:3)

Now find determinant of J and find paramters that make this determinant equal to zero:

Determinant(J) = -cos(q2)*(d3 + a2)^2 = 0

q2= pi/2 or 3pi/2

d3= -a2

for q2=pi/2 :

when first and second link have the same axis this makes singularity ; makes robot loses 1 DOF

for q2=3pi/2 :

same as prvious case but this time in opposite direction so 2 links interfere … to be possible physically we need to make them avoid collision otherwise it can`t be implemented in real life

for d3= -a2:

this means that the end effector position lie on the axes of the first link so it losses 1 DOF

5- Compute the velocity of the tool frame:

Position of Joints:

$$q1 = \sin t$$

$$q2 = \cos 2t$$

$$d3 = \sin 3t$$

Velocity of Joints:

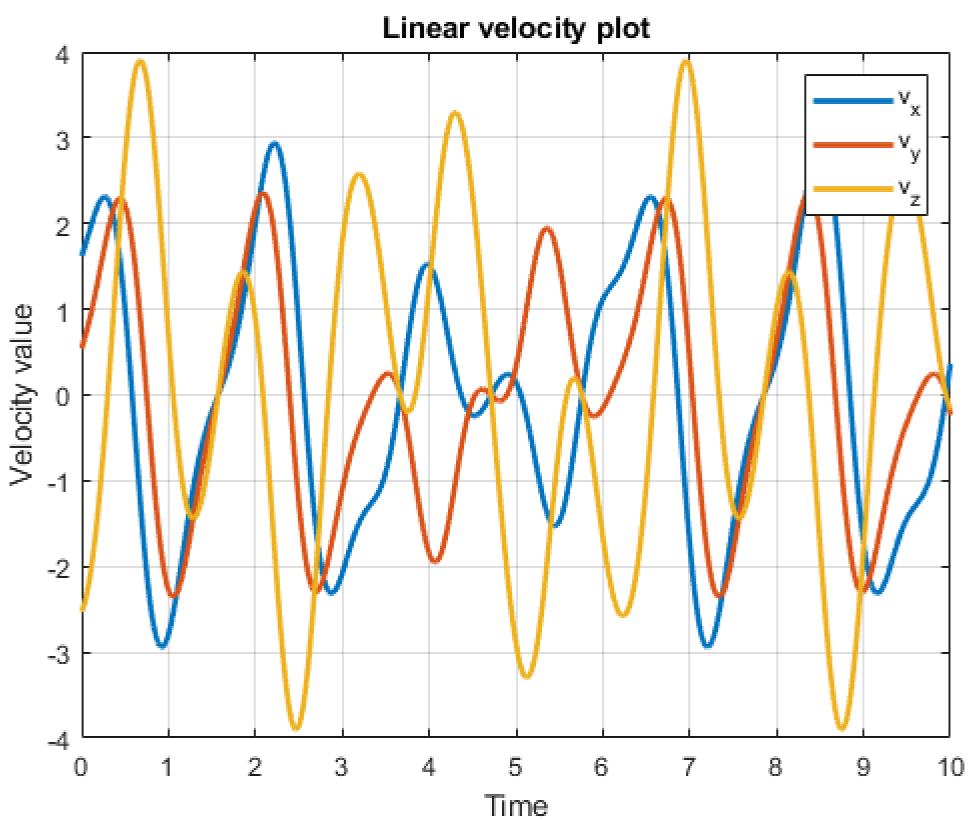
$$\dot{q}1 = \cos t$$

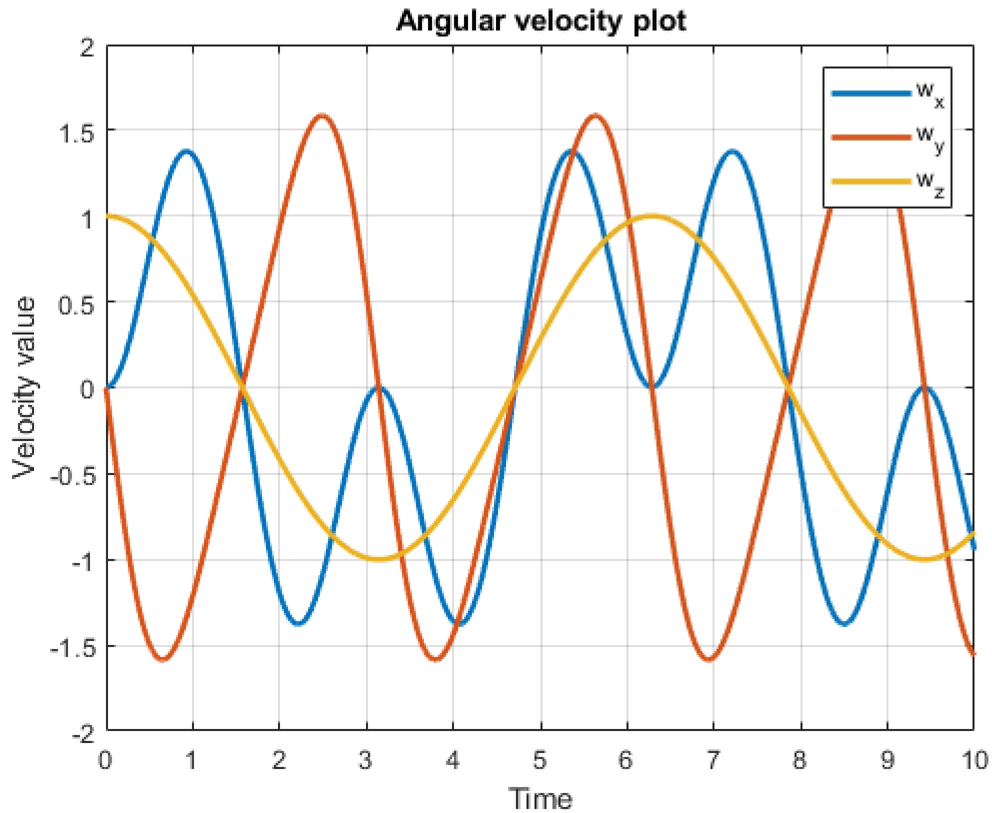
$$\dot{q}2 = -2\sin 2t$$

$$\dot{d}3 = 3\cos 3t$$

Velocity calculation for end effector:

$$\begin{bmatrix} V_{3x1} \\ W_{3x1} \end{bmatrix} = J_{6x3}\ X \begin{bmatrix} \dot{q}1 \\ \dot{q}2 \\ \dot{d}3 \end{bmatrix}$$





6- Here's the Links to my GitHub:

https://github.com/Mohamed-Moustafa/DONRS_HW3.git