```
clc, clear all, close all
```

Laboratorio 3 - Modelo diferencial de primer orden

Camilo Ernesto Campo Pacheco - Nicolas Pulido Gerena - Manuel Alejandro Rojas Cubillos

El repositorio de Github lo pueden encontrar en el siguiente Link

El siguiente documento muestra el trabajo realizado para el Laboratorio 3 de la materia Robótica donde se busca entender y simular el modelo diferencial de primer orden de un robot industrial asignado teniendo en cuenta sus características.

Para este laboratorio se tiene ya planteado desde laboratorios anteriores el analisis geometrico del robot **YUANDA** (Robot articulado y colaborativo de 6 ejes).

Robot Yuanda

i	α_(i-1)	a_(i-1)	d_i	θ_i	Offset
1	0	0	L1	θ1	0
2	-pi/2	0	L2	θ2	0
3	0	L3	0	θ3	0
4	0	L5	-L4-L6	θ4	0
5	pi/2	0	L7	θ5	pi/2
6	pi/2	0	L8	θ6	0

Tabla de parametros DHmod

Modelo diferencial de primer Orden

Considerando que para que el robot ejecute los movimientos, se desea establecer una relaci´on entre las velocidades del efector final y las articulaciones.

1.Con los valores numéricos de longitudes y desplazamientos de los eslabones del robot asignado obtenga el Jacobiano (como una matriz de valores numéricos, NO simbólicos) en función de los ángulos de articulación.

A partir de la tabla de parametros DHmod se utilizan los parametros de DH estandar de la siguiente forma:

```
L1=5; L2=5; L3=30; L4=5; L5=45; L6=5; L7=5; L8=10; %Dimensiones en cm
ws = [-600 1000 -1000 1000 -1000 1000]/10;

plot_options = {'workspace',ws,'scale',.5,'view',[-5 25], 'tilesize',2,  'ortho', 'lightpos',[2

L(1) = Link('revolute'   ,'alpha',    0,    'a',  0,   'd',    L1,      'offset',     0,    'q
L(2) = Link('revolute'   ,'alpha',    -pi/2,'a',  0,   'd',    L2,      'offset',     0,    'q
L(3) = Link('revolute'   ,'alpha',    0,    'a',  L3,  'd',    0,       'offset',     0,    'q
L(4) = Link('revolute'   ,'alpha',    0,    'a',  L5,  'd',    -L4-L6 , 'offset',     0,    'q
L(5) = Link('revolute'   ,'alpha',    pi/2, 'a',  0,   'd',    L7,      'offset',     pi/2, 'q
L(6) = Link('revolute'   ,'alpha',    pi/2, 'a',  0,   'd',    L8,      'offset',     0,    'q

Robot = SerialLink(L,'name','YUANDA','plotopt',plot_options)
```

```
Robot =

YUANDA (6 axis, RRRRRR, modDH, fastRNE)

+---+-----------+-----------+-----------+-----------+-----------+
| j |     theta |         d |         a |     alpha |    offset |
+---+-----------+-----------+-----------+-----------+-----------+
|  1|         q1|          5|          0|          0|          0|
|  2|         q2|          5|          0|     -1.571|          0|
|  3|         q3|          0|         30|          0|          0|
|  4|         q4|        -10|         45|          0|          0|
|  5|         q5|          5|          0|      1.571|      1.571|
|  6|         q6|         10|          0|      1.571|          0|
+---+-----------+-----------+-----------+-----------+-----------+

grav =    0  base = 1  0  0  0   tool =  1  0  0  0
          0         0  1  0  0           0  1  0  0
       9.81         0  0  1  0           0  0  1  0
                    0  0  0  1           0  0  0  1
```

```
q = [-47*pi/120 0 0 -pi/2 -pi/2 0];

A_0_1 = L(1).A(q(1))
```

```
A_0_1 = 4×4
    0.3338    0.9426         0         0
   -0.9426    0.3338         0         0
         0         0    1.0000    5.0000
         0         0         0    1.0000
```

```
A_1_2 = L(2).A(q(2))
```

```
A_1_2 = 4×4
    1.0000         0         0         0
         0    0.0000    1.0000    5.0000
         0   -1.0000    0.0000    0.0000
```

```
         0         0         0    1.0000
```

```
A_2_3 = L(3).A(q(3))
```

```
A_2_3 = 4×4
     1     0     0    30
     0     1     0     0
     0     0     1     0
     0     0     0     1
```

```
A_3_4 = L(4).A(q(4))
```

```
A_3_4 = 4×4
    0.0000    1.0000         0   45.0000
   -1.0000    0.0000         0         0
         0         0    1.0000  -10.0000
         0         0         0    1.0000
```

```
A_4_5 = L(5).A(q(5))
```

```
A_4_5 = 4×4
    1.0000         0         0         0
         0    0.0000   -1.0000   -5.0000
         0    1.0000    0.0000    0.0000
         0         0         0    1.0000
```

```
A_5_6 = L(6).A(q(6))
```

```
A_5_6 = 4×4
    1.0000         0         0         0
         0    0.0000   -1.0000  -10.0000
         0    1.0000    0.0000    0.0000
         0         0         0    1.0000
```

```
T_0_1 = A_0_1 * eye(4)
```

```
T_0_1 = 4×4
    0.3338    0.9426         0         0
   -0.9426    0.3338         0         0
         0         0    1.0000    5.0000
         0         0         0    1.0000
```

```
T_0_2 = A_0_1 * A_1_2 * eye(4)
```

```
T_0_2 = 4×4
    0.3338    0.0000    0.9426    4.7132
   -0.9426    0.0000    0.3338    1.6690
         0   -1.0000    0.0000    5.0000
         0         0         0    1.0000
```

```
T_0_3 = A_0_1 * A_1_2 * A_2_3 * eye(4)
```

```
T_0_3 = 4×4
    0.3338    0.0000    0.9426   14.7274
   -0.9426    0.0000    0.3338  -26.6102
         0   -1.0000    0.0000    5.0000
         0         0         0    1.0000
```

```
T_0_4 = A_0_1 * A_1_2 * A_2_3 * A_3_4 * eye(4)
```

```
T_0_4 = 4×4
   -0.0000    0.3338    0.9426   20.3223
   -0.0000   -0.9426    0.3338  -72.3671
    1.0000   -0.0000    0.0000    5.0000
         0         0         0    1.0000
```

```
T_0_5 = A_0_1 * A_1_2 * A_2_3 * A_3_4 * A_4_5 * eye(4)
```

```
T_0_5 = 4×4
   -0.0000    0.9426   -0.3338   18.6533
   -0.0000    0.3338    0.9426  -67.6539
    1.0000    0.0000    0.0000    5.0000
         0         0         0    1.0000
```

```
T_0_6 = A_0_1 * A_1_2 * A_2_3 * A_3_4 * A_4_5 * A_5_6 * eye(4)
```

```
T_0_6 = 4×4
   -0.0000   -0.3338   -0.9426    9.2269
   -0.0000    0.9426   -0.3338  -70.9920
    1.0000    0.0000   -0.0000    5.0000
         0         0         0    1.0000
```

```
Z_0 = [0; 0; 1]
```

```
Z_0 = 3×1
     0
     0
     1
```

```
Z_1 = [T_0_1(1,3); T_0_1(2,3); T_0_1(3,3)]
```

```
Z_1 = 3×1
     0
     0
     1
```

```
Z_2 = [T_0_2(1,3); T_0_2(2,3); T_0_2(3,3)]
```

```
Z_2 = 3×1
    0.9426
    0.3338
    0.0000
```

```
Z_3 = [T_0_3(1,3); T_0_3(2,3); T_0_3(3,3)]
```

```
Z_3 = 3×1
    0.9426
    0.3338
    0.0000
```

```
Z_4 = [T_0_4(1,3); T_0_4(2,3); T_0_4(3,3)]
```

```
Z_4 = 3×1
    0.9426
    0.3338
    0.0000
```

```
Z_5 = [T_0_5(1,3); T_0_5(2,3); T_0_5(3,3)]
```

```
Z_5 = 3×1
```

```
   -0.3338
    0.9426
    0.0000
```

```
T_0 = [0; 0; 0]
```

```
T_0 = 3×1
     0
     0
     0
```

```
T_1 = [T_0_1(1,4); T_0_1(2,4); T_0_1(3,4)]
```

```
T_1 = 3×1
     0
     0
     5
```

```
T_2 = [T_0_2(1,4); T_0_2(2,4); T_0_2(3,4)]
```

```
T_2 = 3×1
    4.7132
    1.6690
    5.0000
```

```
T_3 = [T_0_3(1,4); T_0_3(2,4); T_0_3(3,4)]
```

```
T_3 = 3×1
   14.7274
  -26.6102
    5.0000
```

```
T_4 = [T_0_4(1,4); T_0_4(2,4); T_0_4(3,4)]
```

```
T_4 = 3×1
   20.3223
  -72.3671
    5.0000
```

```
T_5 = [T_0_5(1,4); T_0_5(2,4); T_0_5(3,4)]
```

```
T_5 = 3×1
   18.6533
  -67.6539
    5.0000
```

```
T_6 = [T_0_6(1,4); T_0_6(2,4); T_0_6(3,4)]
```

```
T_6 = 3×1
    9.2269
  -70.9920
    5.0000
```

```
J = [cross(Z_0,(T_6-T_0)) cross(Z_1,(T_6-T_1)) cross(Z_2,(T_6-T_2))...
    cross(Z_3,(T_6-T_3)) cross(Z_4,(T_6-T_4)) cross(Z_5,(T_6-T_5));
    Z_0 Z_1 Z_2 Z_3 Z_4 Z_5]
```

```
J = 6×6
```

```
   70.9920   70.9920    0.0000    0.0000   -0.0000   -0.0000
    9.2269    9.2269    0.0000    0.0000    0.0000   -0.0000
         0         0  -70.0000  -40.0000    5.0000   10.0000
         0         0    0.9426    0.9426    0.9426   -0.3338
         0         0    0.3338    0.3338    0.3338    0.9426
    1.0000    1.0000    0.0000    0.0000    0.0000    0.0000
```

El valor de "J" visto anteriormente, corresponde al Jacobiano.

2. Para una postura de su elección dentro del espacio diestro del robot obtenga las velocidades de articulación para:

$$V_H = \begin{bmatrix} 100 \\ 200 \\ 50 \end{bmatrix} mm/s \qquad \omega_H = \begin{bmatrix} 5 \\ 10 \\ -5 \end{bmatrix} rad/s$$

Para el inverso del Jacobiano se utiliza el siguiente comando:

```
Jinv = pinv(J)
```

```
Jinv = 6×6
    0.0069    0.0009    0.0000    0.0000    0.0000    0.0001
    0.0069    0.0009    0.0000    0.0000    0.0000    0.0001
   -0.0000    0.0000   -0.0123   -0.1320    0.0836    0.0000
    0.0000   -0.0000   -0.0018    0.2505    0.1073    0.0000
    0.0000   -0.0000    0.0140    0.8241    0.1429    0.0000
   -0.0000    0.0000    0.0000   -0.3338    0.9426    0.0000
```

Obteniendo asi:

```
V_H = [100;200;50]
```

```
V_H = 3×1
   100
   200
    50
```

```
w_H = [5;10;-5]
```

```
w_H = 3×1
     5
    10
    -5
```

```
vector = [V_H;w_H]
```

```
vector = 6×1
   100
   200
    50
     5
    10
    -5
```

```
q_punto = Jinv*vector
```

```
q_punto = 6×1
    0.8720
```

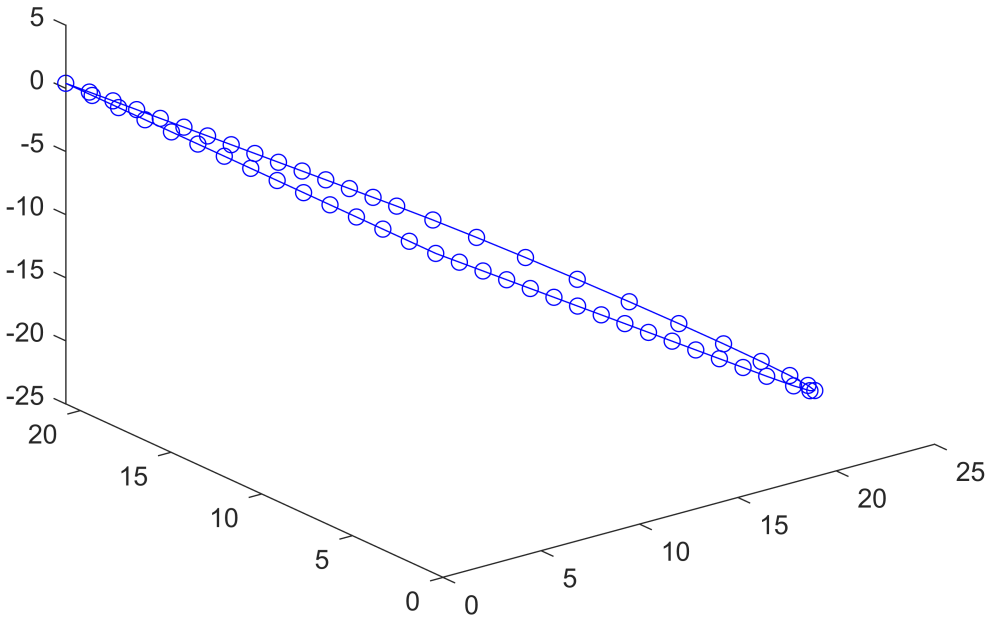
```
    0.8720
   -0.4383
    2.2378
    6.2518
    7.7574
```

Integración

Ahora con la ayuda de los algoritmos desarrollados y de la GUI construida.

1. Ubique la ruta seleccionada con la orientación indicada dentro del espacio diestro del robot.

```
Horizontal = 85; %valor en cm
L = 0.4*Horizontal;

r = 0.3*L;
n = 15;

t1 = [linspace(0,L-r,n)' zeros(n,2)];
theta = linspace(-pi/2,pi/2, n);
cx = (L-r);
cy = r;
cz = 0;

t2 = [cx+r*cos(theta)' cy+r*sin(theta)' cz+0.*theta'];
t3 = [t2(end,:);linspace(L-r,0,n)' 2*r.*ones(n,1) zeros(n,1)];
t4 = [t3(end,:);zeros(n,1) linspace(2*r,0,n)' zeros(n,1)];

ruta = [t1; t2; t3; t4];

clf
R0_t = roty(45,'deg'); %%Definición del plano-vector [-1, 0, 1]
p0_t = [0 0.4 0.4];
ruta_ = 0.*ruta;
for i=1:length(ruta(:,1))
    ruta_(i,:) = R0_t*ruta(i,:)'+p0_t';
end

figure (1)
plot3(ruta_(:,1),ruta_(:,2),ruta_(:,3),'-bo')
```



2. Calcule las configuraciones correspondientes a cada viapoint. Presente una gráfica de cada ángulo de articulación al recorrer la ruta.

Se crea el robot utilizando RST:

```
robot = rigidBodyTree;
body1 = rigidBody('body1');
body2 = rigidBody('body2');
body3 = rigidBody('body3');
body4 = rigidBody('body4');
body5 = rigidBody('body5');
body6 = rigidBody('body6');
bodyEndEffector=rigidBody('endeffector');
jnt1 = rigidBodyJoint('jnt1','revolute');
jnt2 = rigidBodyJoint('jnt2','revolute');
jnt3 = rigidBodyJoint('jnt3','revolute');
jnt4 = rigidBodyJoint('jnt4','revolute');
jnt5 = rigidBodyJoint('jnt5','revolute');
jnt6 = rigidBodyJoint('jnt6','revolute');
%body1
tform1 = transl(0, 0, 5);
setFixedTransform(jnt1,tform1);
body1.Joint = jnt1;
addBody(robot,body1,'base')

%body2
tform2 = transl(0, 0, 5)*troty(-pi/2);
setFixedTransform(jnt2,tform2);
```

```
body2.Joint = jnt2;
addBody(robot,body2,'body1')

%body3
tform3 = transl(0, 30, 0);
setFixedTransform(jnt3,tform3);
body3.Joint = jnt3;
addBody(robot,body3,'body2')

%body4
tform4 = transl(0, 45, -10);
setFixedTransform(jnt4,tform4);
body4.Joint = jnt4;
addBody(robot,body4,'body3')

%body5
tform5 = transl(0, 0, 5)*troty(pi/2)*trotz(-pi/2);
setFixedTransform(jnt5,tform5);
body5.Joint = jnt5;
addBody(robot,body5,'body4')

%body6
tform6 = transl(0, 0, 10)*troty(pi/2);
setFixedTransform(jnt6,tform6);
body6.Joint = jnt6;
addBody(robot,body6,'body5')

%endEffector
tform7 = trvec2tform([0 0 0]);
setFixedTransform(bodyEndEffector.Joint,tform7);
addBody(robot,bodyEndEffector,'body6')
```

Ahora se realiza la cinematica inversa:

```
ik = inverseKinematics('RigidBodyTree', robot);
weights = [0.25 0.25 0.25 1 1 1];
initialguess = homeConfiguration(robot);
robot.BodyNames
```

```
ans = 1×7 cell
'body1'      'body2'      'body3'      'body4'      'body5'      'body6'      'e···
```

```
viapoints = length(ruta(:,1));
conf_ruta = zeros(viapoints,6);
for i=1:viapoints
    tic
    if i>1
        initialguess = configSoln;
    end
    p0_t = ruta_(i,:);
    R0_t = rotx(-135,'deg');
    MTHpose = [R0_t p0_t'; 0 0 0 1];
    [configSoln,solnInfo] = ik('endeffector',MTHpose,weights,initialguess);
```

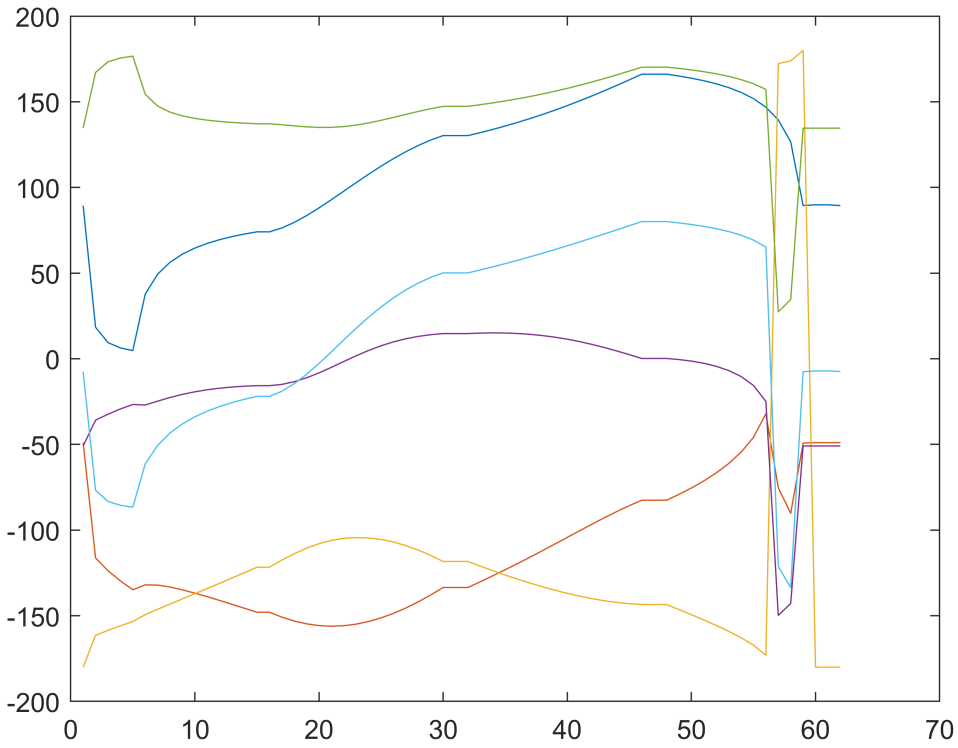
```
    conf_ruta(i,:) = [configSoln(1).JointPosition configSoln(2).JointPosition configSoln(3).Joi
        configSoln(4).JointPosition configSoln(5).JointPosition configSoln(6).JointPosition];
    rutaxd(i,:) = configSoln';
    disp(strcat(num2str(i),"/",num2str(viapoints)))
    toc
end
```

```
1/62
Elapsed time is 9.042718 seconds.
2/62
Elapsed time is 5.891750 seconds.
3/62
Elapsed time is 5.443501 seconds.
4/62
Elapsed time is 5.595543 seconds.
5/62
Elapsed time is 4.270580 seconds.
6/62
Elapsed time is 0.161496 seconds.
7/62
Elapsed time is 0.208221 seconds.
8/62
Elapsed time is 0.270701 seconds.
9/62
Elapsed time is 0.305262 seconds.
10/62
Elapsed time is 0.190091 seconds.
11/62
Elapsed time is 0.237186 seconds.
12/62
Elapsed time is 0.457756 seconds.
13/62
Elapsed time is 0.316225 seconds.
14/62
Elapsed time is 0.246142 seconds.
15/62
Elapsed time is 0.535361 seconds.
16/62
Elapsed time is 0.046488 seconds.
17/62
Elapsed time is 0.263406 seconds.
18/62
Elapsed time is 0.370931 seconds.
19/62
Elapsed time is 0.313211 seconds.
20/62
Elapsed time is 0.237715 seconds.
21/62
Elapsed time is 0.266663 seconds.
22/62
Elapsed time is 0.272716 seconds.
23/62
Elapsed time is 0.256918 seconds.
24/62
Elapsed time is 0.233263 seconds.
25/62
Elapsed time is 0.274307 seconds.
26/62
Elapsed time is 0.259462 seconds.
27/62
Elapsed time is 0.393116 seconds.
28/62
Elapsed time is 0.400306 seconds.
```

```
29/62
Elapsed time is 0.339093 seconds.
30/62
Elapsed time is 0.322903 seconds.
31/62
Elapsed time is 1.457218 seconds.
32/62
Elapsed time is 0.011201 seconds.
33/62
Elapsed time is 0.303961 seconds.
34/62
Elapsed time is 0.278128 seconds.
35/62
Elapsed time is 0.268598 seconds.
36/62
Elapsed time is 0.219580 seconds.
37/62
Elapsed time is 0.264320 seconds.
38/62
Elapsed time is 0.234948 seconds.
39/62
Elapsed time is 0.204092 seconds.
40/62
Elapsed time is 0.343411 seconds.
41/62
Elapsed time is 0.252105 seconds.
42/62
Elapsed time is 0.286276 seconds.
43/62
Elapsed time is 0.179289 seconds.
44/62
Elapsed time is 0.311190 seconds.
45/62
Elapsed time is 0.259029 seconds.
46/62
Elapsed time is 0.254210 seconds.
47/62
Elapsed time is 0.008522 seconds.
48/62
Elapsed time is 0.008501 seconds.
49/62
Elapsed time is 0.280260 seconds.
50/62
Elapsed time is 0.246067 seconds.
51/62
Elapsed time is 0.369932 seconds.
52/62
Elapsed time is 0.275264 seconds.
53/62
Elapsed time is 0.243974 seconds.
54/62
Elapsed time is 0.203411 seconds.
55/62
Elapsed time is 0.197005 seconds.
56/62
Elapsed time is 0.213042 seconds.
57/62
Elapsed time is 0.723858 seconds.
58/62
Elapsed time is 0.153015 seconds.
59/62
Elapsed time is 7.286962 seconds.
60/62
Elapsed time is 5.965414 seconds.
```

```
61/62
Elapsed time is 6.585041 seconds.
62/62
Elapsed time is 6.741381 seconds.
```

```
figure (2)
plot(conf_ruta(:,1)*180/pi)
hold on
plot(conf_ruta(:,2)*180/pi)
hold on
plot(conf_ruta(:,3)*180/pi)
hold on
plot(conf_ruta(:,4)*180/pi)
hold on
plot(conf_ruta(:,5)*180/pi)
hold on
plot(conf_ruta(:,6)*180/pi)
hold off
```



Para la velocidad fija en la trayectoria (500mm/s):

```
Jacobianos = cell(1,viapoints);
for i=1:viapoints
    Jacobianos{i} = geometricJacobian(robot,rutaxd(i,:),'endeffector');
end

tiempos = zeros(61,1);
distancia_tot = 0;
```

```
distancias = zeros(61,3);
velocidad = 0.5;
for j=1:viapoints-1
    norma = norm(ruta_(j+1,:)-ruta_(j,:));
    distancia_tot = distancia_tot + norma;
    if norma ~= 0
        distancias(j,:) = (ruta_(j+1,:)-ruta_(j,:));
        tiempos(j) = norma/velocidad;
    else
        distancias(j,:) = [0 0 0];
    end
end
```

Para la velocidad lineal instantánea:

```
v = zeros(61,3);
for k=1:viapoints-1
    if(tiempos(k) ~= 0)
        v(k,:) = (distancias(k)/tiempos(k));
    else
        v(k,:) = [0 0 0];
    end
end
v = [[0 0 0]; v];
w = zeros(62,3);
vw = [v, w];
q_punto = zeros(6,62);

for i=1:viapoints
    q_punto(:,i) = inv(Jacobianos{i})*vw(i,:)';
end
```

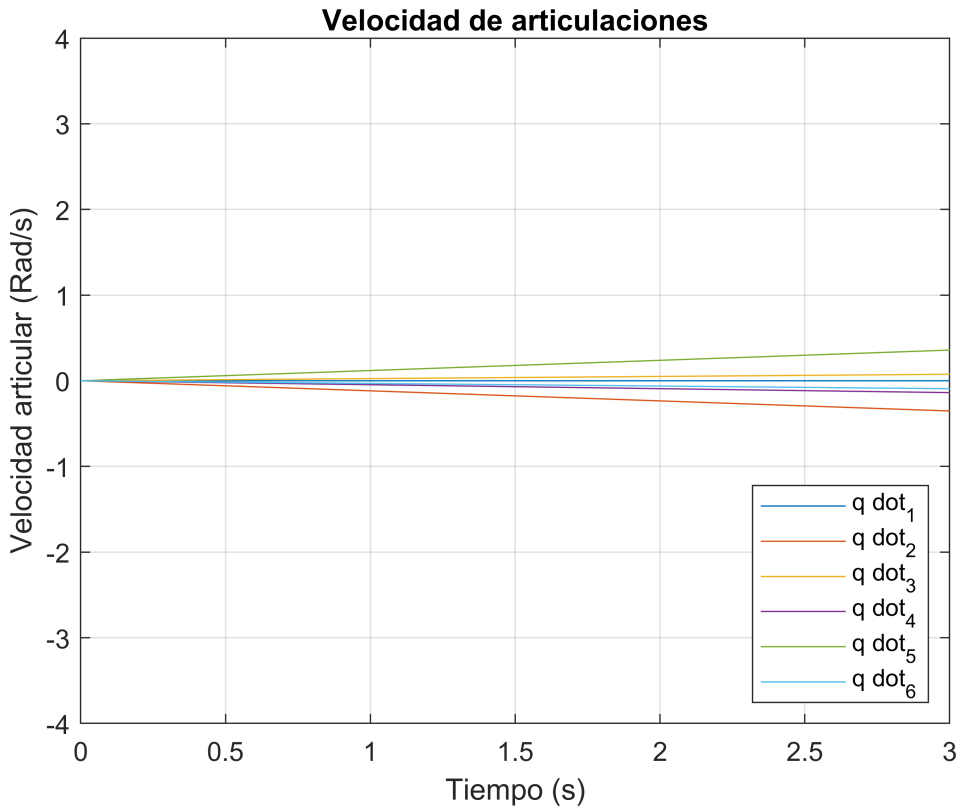
```
Warning: Matrix is close to singular or badly scaled. Results may be inaccurate. RCOND =  6.599682e-18.
```

```
t = zeros(62,1);
for i = 2 : viapoints
    t(i) = t(i-1) + tiempos(i-1);
end
```

Graficando:

```
clf
for i=1:6
    plot(t,q_punto(i,:), 'DisplayName',['q dot_',num2str(i)])
    hold on
end
legend('Location',"southeast")
grid on
hold off
title('Velocidad de articulaciones')
xlim([0.00 3.00])
ylim([-4.00 4.00])
xlabel('Tiempo (s)')
```

```
ylabel('Velocidad articular (Rad/s)')
```



Para la animación del Robot, se utiliza el siguiente codigo en comentario para evitar errores de compilación (Se cuenta como anexo el video en Github donde se presenta la animación):

```
% fps=60;r=rateControl(fps);
% show(robot,rutaxd(1,:))
% ax=gca;
% hold on
% plot3(ax,ruta_(:,1),ruta_(:,2),ruta_(:,3),'-bo');
% hold on
% while true
% for i=1:viapoints
%     show(robot,rutaxd(i,:),'PreservePlot',false);
%     drawnow
%     waitfor(r);
% end
% end
```

El repositorio de Github lo pueden encontrar en el siguiente Link