

Informe de Trabajo Práctico N°8

Planificación y Generación de Trayectorias

Robótica I

Ingeniería en Mecatrónica

Facultad de Ingeniería - UNCUIYO

Alumno: Juan Manuel BORQUEZ PEREZ

Legajo: 13567



UNCUIYO
UNIVERSIDAD
NACIONAL DE CUYO



**FACULTAD
DE INGENIERÍA**

1. Ejercicio 1: Generación de trayectoria entre 2 puntos articulares

1.1. Interpolación.

La interpolación con “jtraj” es interpolación quintuple en el espacio articular. Se indica la posición articular inicial y final y un vector de tiempos (como en este caso) o un número de puntos en la discretización (como en el ejercicio siguiente). Cuando se da un vector de tiempos, la trayectoria se da evaluada en esos instantes y cuando se indica un número de puntos, la trayectoria se da evaluada en intervalos equi-espaciados en un tiempo unitario.

```
1      % Inciso 1
2      q0 = [0      -pi/2      0      0      0 0]; % Posicion inicial
3      q1 = [-pi/3  pi/10 -pi/5  pi/2  pi/4 0]; % Posicion final
4      ti = 0; % Instante inicial
5      tf = 3; % Instante final
6      dt = 0.1; % Paso de tiempo
7      t = ti:dt:tf; % 3 seg. y una decima de
           paso
8      [Q, QD, QDD] = jtraj(q0, q1, t); % Trayectoria interpolada
```

La trayectoria interpolada “Q” es de 31 X 6 y tiene en filas la sucesión de posiciones articulares entre la inicial y la final, incluyéndolas. Concuera con el límite y paso de tiempo. Esto es, en 3 segundos hay 30 decimas y al considerar el instante inicial se tienen 31 puntos.

Las matrices “QD” y “QDD” son de velocidad y aceleración articular respectivamente y tienen las mismas dimensiones que “Q”.

1.2. Animación

Se muestra la trayectoria trazada en fig. 1

```
1      % Inciso 2
2      figure(1);
3      for q = Q'
4          fanuc.animate(q');
5          pause(dt);
6      end
```

1.3. Grafico de las variables articulares.

```
1      %% Inciso 3
2      figure(2);
3      qplot(t', Q);
```

Se puede ver en fig. 4 que el movimiento de los ejes es simultáneo y síncrono, es decir, que los ejes comienzan su movimiento en el mismo instante de tiempo y también lo terminan en el mismo instante de tiempo.

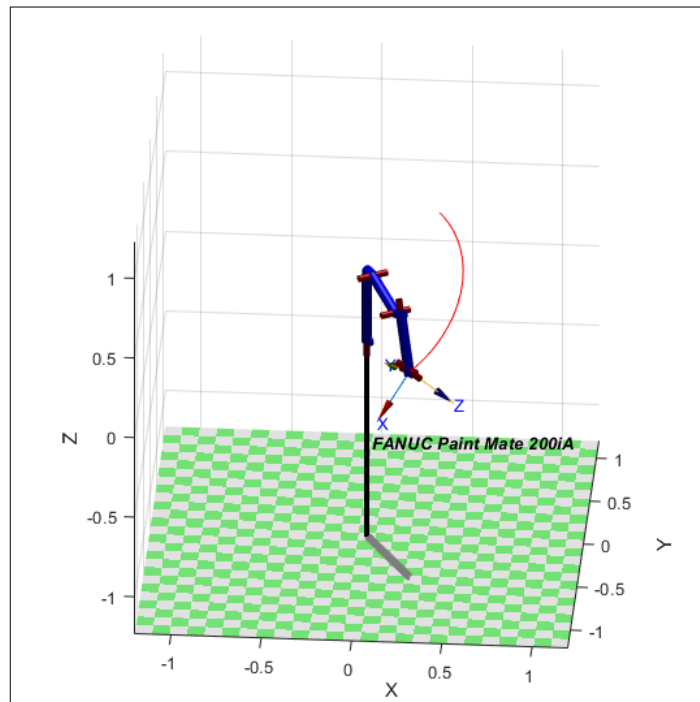


Figura 1: Animación Fanuc.

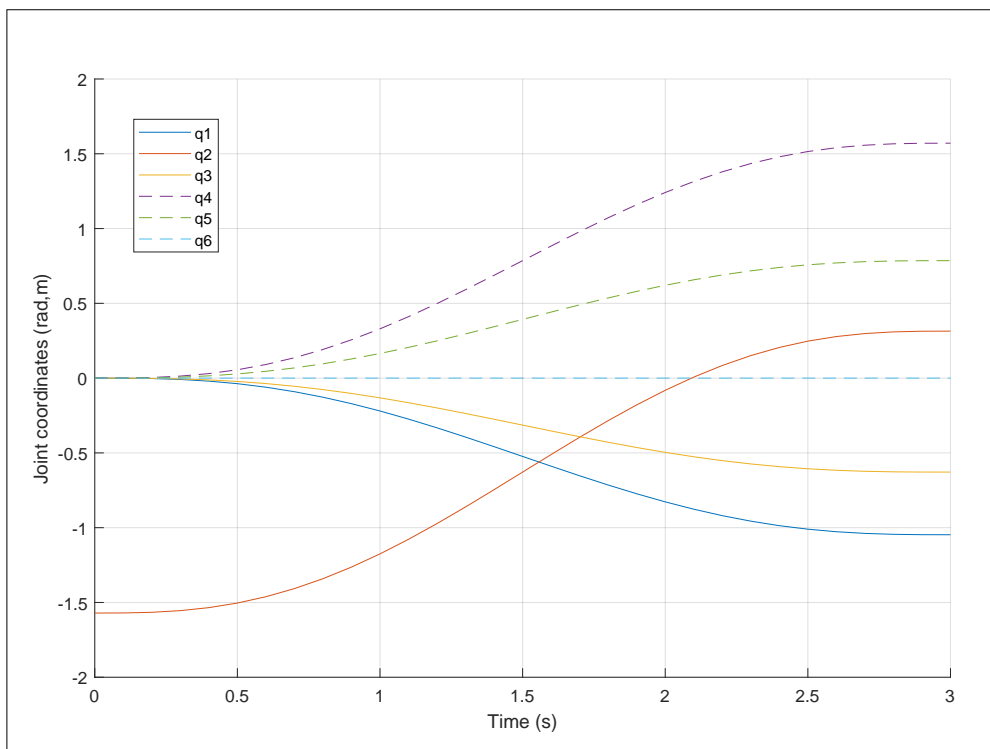


Figura 2: qplot Fanuc.

2. Ejercicio 2: Generación de Trayectorias entre puntos cartesianos. Interpolación articular.

2.1. Generación

```
1 %% Inciso 1
2 p1 = [0.0 0.0 0.95]; % Posicion incial
3 p2 = [0.4 0.0 0.95]; % Posicion final
4 qq = [0 -pi/2 -pi/4 0 pi/4 0]; % Posicion articular que da la
   Orientacion
5 R = fanuc.fkine(qq); % Matriz homog. que da la
   Orientacion
6
7 T1 = R; T1.t = p1; % Postura inicial
8 T2 = R; T2.t = p2; % Postura final
9
10 q1 = fanuc.ikine(T1, 'q0', qq, 'mask', ones(1, 6));
11 q2 = fanuc.ikine(T2, 'q0', qq, 'mask', ones(1, 6));
12
13 Q = jtraj(q1, q2, 100);
```

Primero se hace la cinemática directa con “qq”. De la matriz “R” obtenida, solo se cambia el vector de traslación por “p1” para dar “T1” (postura inicial) y por “p2” para dar “T2” (postura final) conservando la sub-matriz de rotación. Luego se usa cinemática inversa para obtener las posiciones articulares correspondientes “q1” y “q2”, con las que se interpola.

2.2. Animación y qplot

```
1 %% Inciso 2
2 figure(1);
3 dt = 0.1;
4 for q = Q'
5     fanuc.animate(q');
6     pause(dt);
7 end
8
9 figure(2);
10 qplot(Q);
```

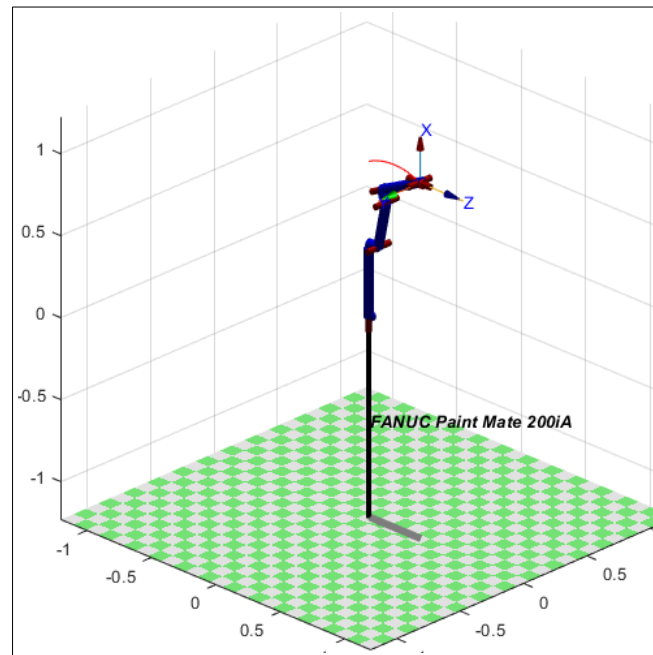


Figura 3: Animación Fanuc.

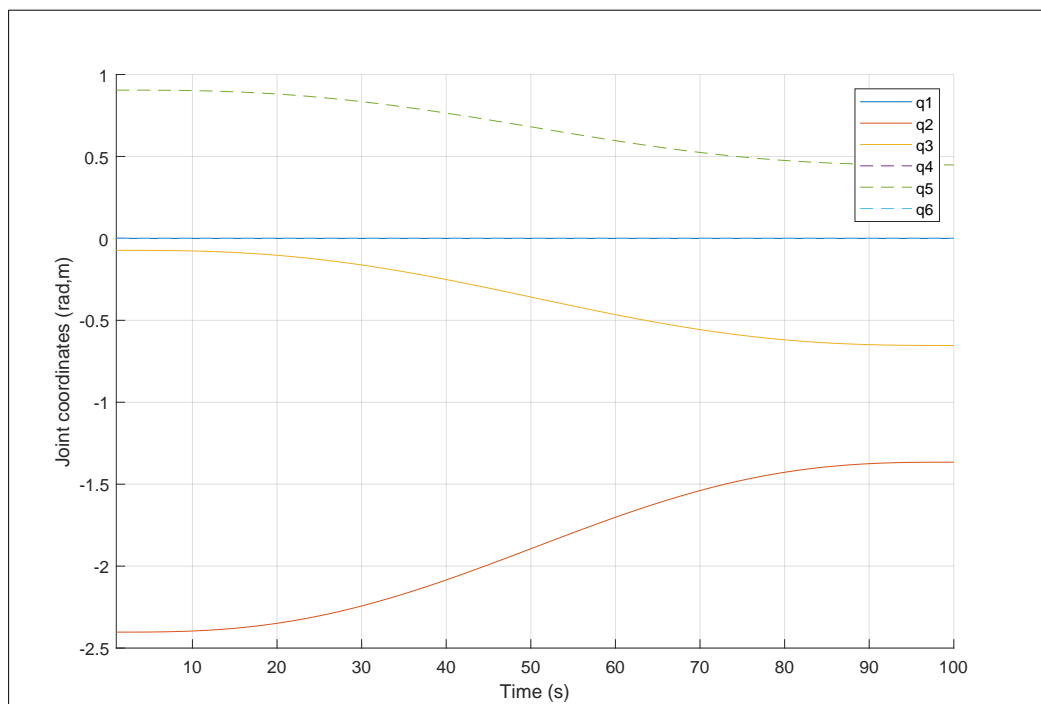


Figura 4: qplot Fanuc.

3. Ejercicio 3: Generación de trayectorias entre 2 puntos cartesianos, interpolación cartesiana

3.1. Interpolación con ctraj

La interpolación con “ctrj” es interpolación cartesiana entre dos posturas en una trayectoria recta siguiendo un perfil de velocidad trapezoidal. Se indican la postura inicial y final con las matrices de transformación homogénea correspondientes y un número de puntos en la discretización o un vector con fracciones de segmento. Cuando se indica un vector de fracciones, la interpolación es evaluada en esas fracciones a lo largo de la trayectoria. El resultado de la interpolación es la secuencia de matrices de transformación homogénea.

```
1 %% Inciso 1
2 p1 = [0.0 0.0 0.95]; % Posicion inicial
3 p2 = [0.4 0.0 0.95]; % Posicion final
4 qq = [0 -pi/2 -pi/4 0 pi/4 0]; % Posicion articular que da la
   Orientacion
5 R = fanuc.fkine(qq); % Matriz homog. que da la
   Orientacion
6
7 T1 = R; T1.t = p1; % Postura inicial
8 T2 = R; T2.t = p2; % Postura final
9 M = 100; % Numero de puntos en la
   interpolacion
10
11 T = ctraj(T1, T2, M); % Trayectoria interpolada
12
13 % Trayectoria en coordenadas articulares
14 Q = fanuc.ikine(T, 'q0', qq, 'mask', ones(1, 6));
```

3.2. Animación y qplot

```
1 %% Inciso 2
2 % Trayectoria en coordenadas articulares
3 Q = fanuc.ikine(T, 'q0', qq, 'mask', ones(1, 6));
4
5 figure(1);
6 fanuc.plot(Q(1, :), 'scale', 0.65, 'jointdiam', 0.65, 'trail', {'r',
   'LineWidth', 0.1});
7 fprintf("\nPresione ENTER para visualizar la animacion del robot.\n
   ");
8 pause;
9
10 dt = 0.1;
11 for q = Q'
12     fanuc.animate(q');
13     pause(dt);
14 end
15
16 figure(2);
17 qplot(Q);
```

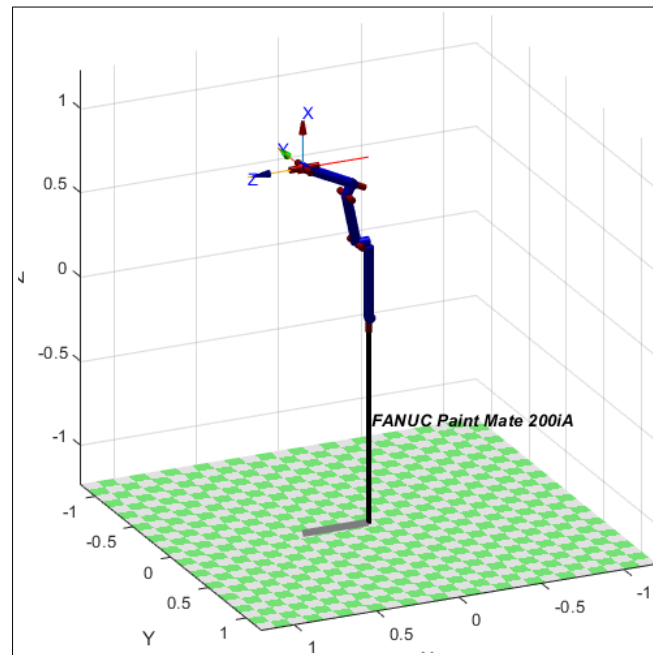


Figura 5: Animación Fanuc.

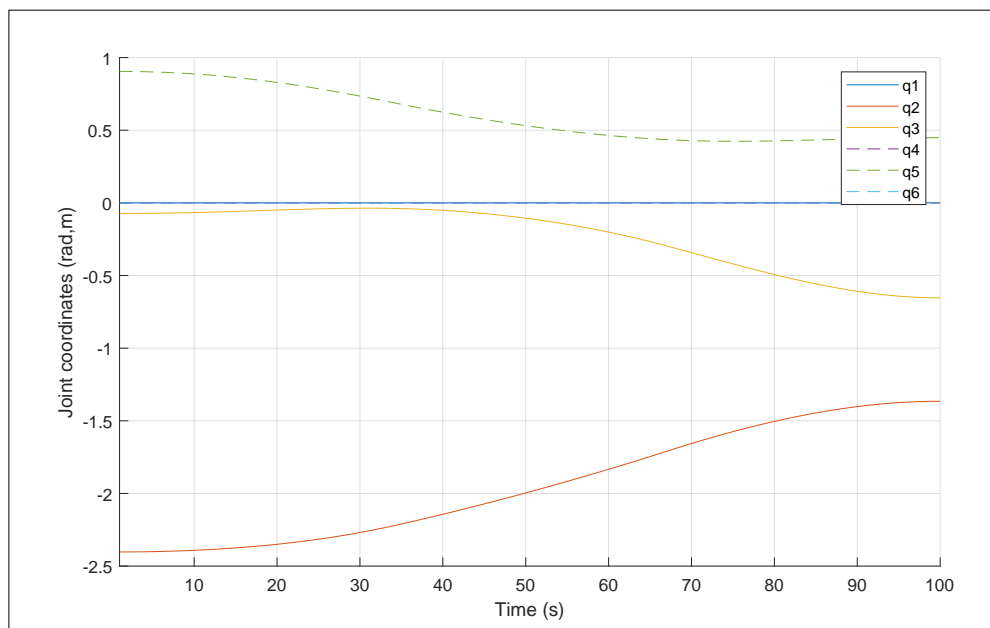


Figura 6: qplot Fanuc.