

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.

```
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
8 % paso
9 Q = jtraj(q0, q1, t); % Trayectoria interpolada
```

Se obtiene una matriz de (31, 6) que tiene en filas la sucesión de posiciones articulares entre la posiciones inicial y final dadas, incluyéndolas. Concuera con el límite y paso de tiempo. Esto es, en 3 segundos hay 30 decimas de segundo y al considerar el instante inicial se tienen 31 puntos.

1.2. Animación

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

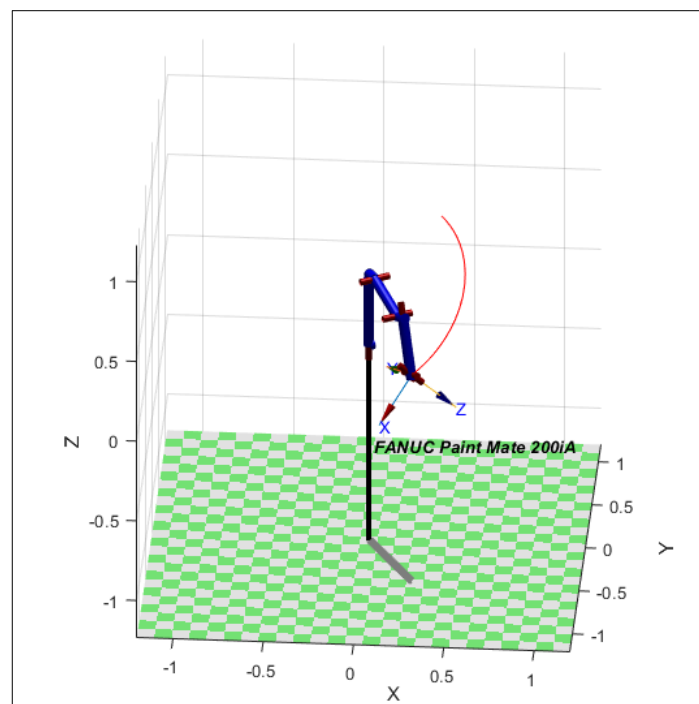


Figura 1: Animación Fanuc.

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.

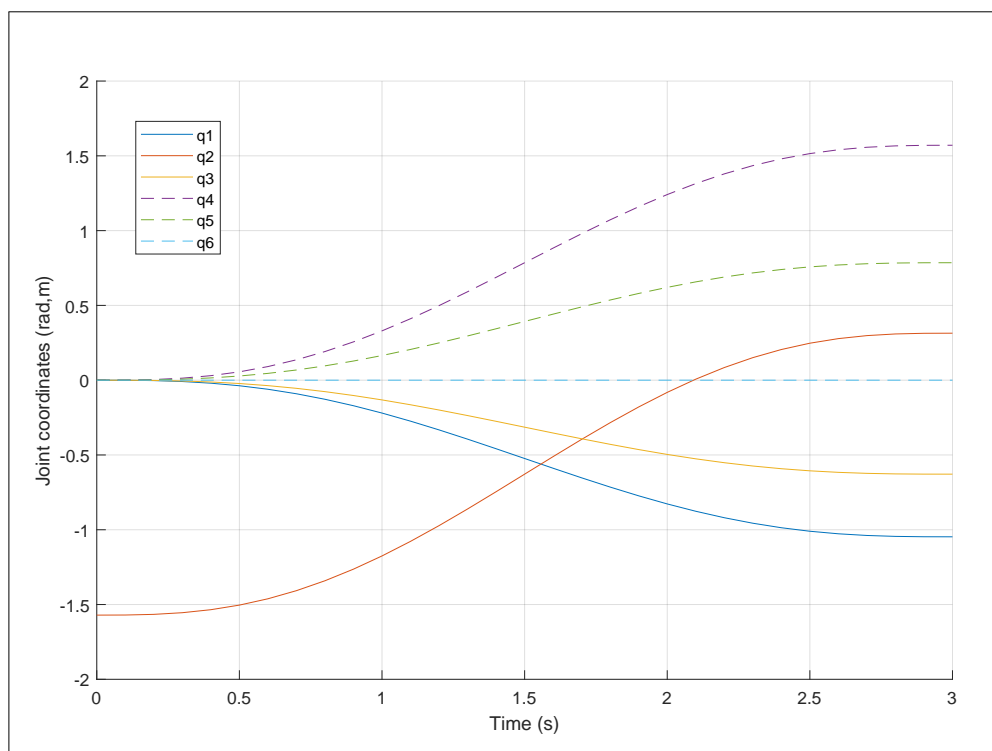


Figura 2: qplot Fanuc.

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 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
```

```
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);
```

Como se puede ver en el código, para definir la postura inicial y final de la trayectoria se hace la cinemática directa del robot. De la matriz obtenida, se cambia el vector de traslación y se conserva la sub-matriz de rotación.

2.2. Animación

```
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);
```

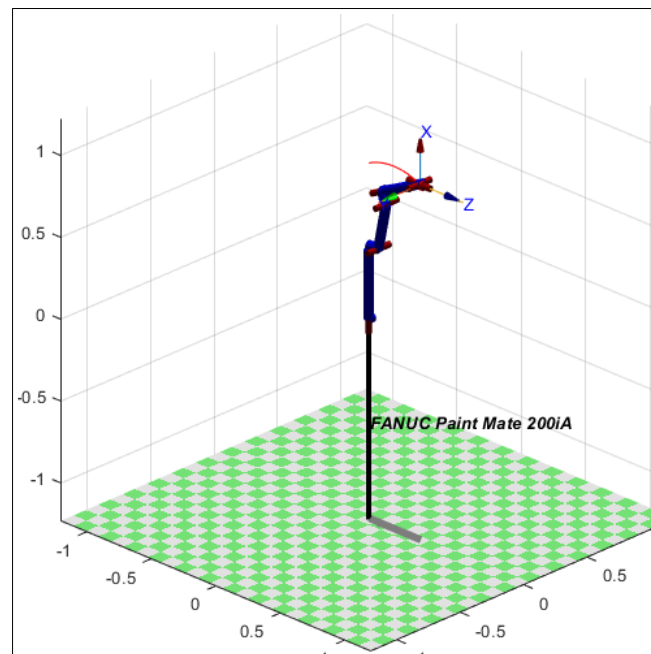


Figura 3: Animación Fanuc.

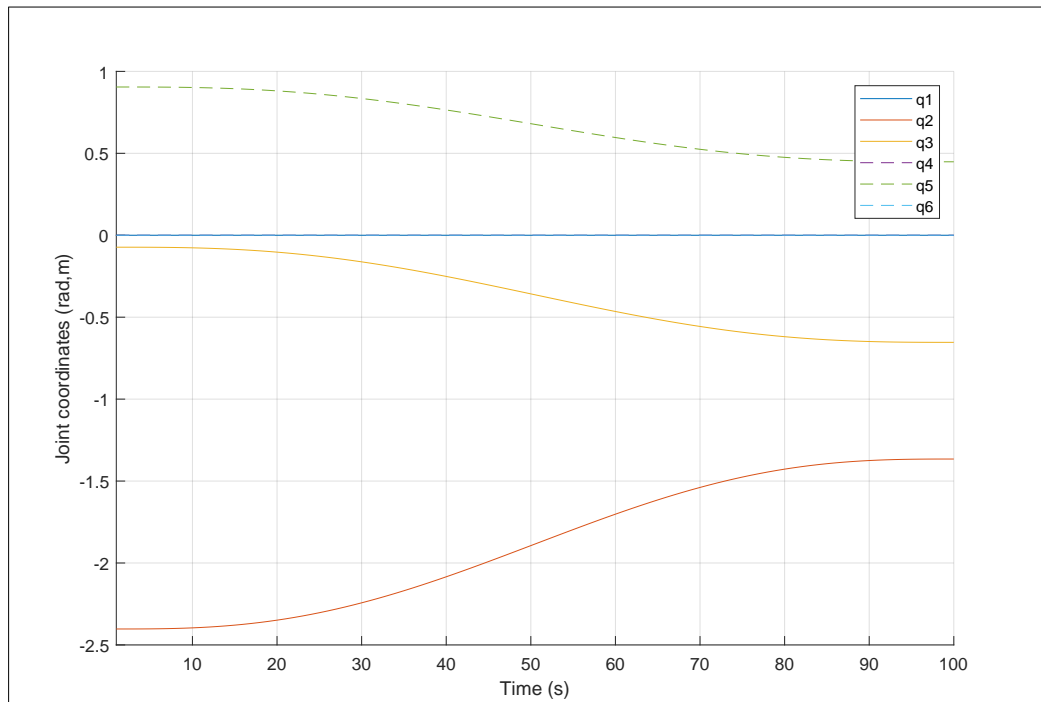


Figura 4: qplot Fanuc.