

ROBÓTICA INDUSTRIAL

---

## Práctica 3: Planificación de la trayectoria del manipulador RR

---

Juan Antonio Aldea Armenteros

20 de junio de 2012

# 1. SIMULACIÓN

## 1.1. TRAYECTORIA COMPLETA

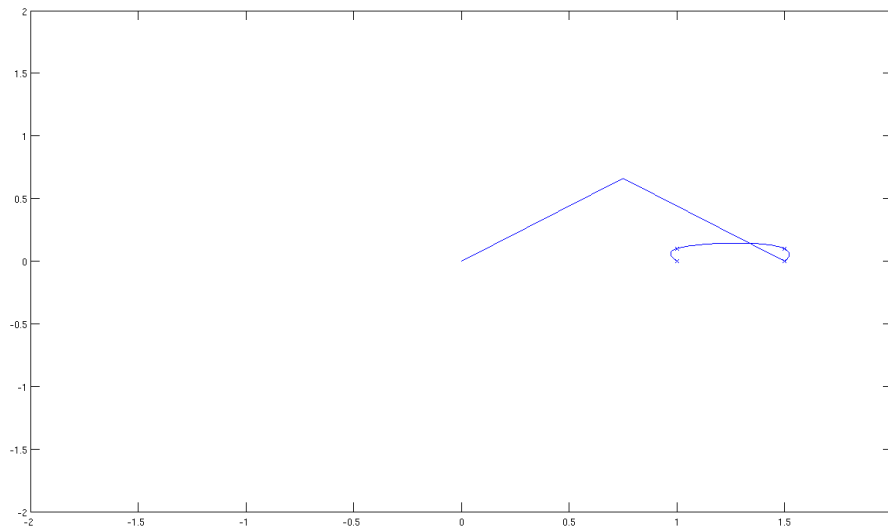


Figura 1.1.1.: Trayectoria Simulada.

La animación puede verse aquí.

## 1.2. CARACTERÍSTICAS TEMPORALES

Como puede apreciarse en las figuras 1.2.1 y 1.2.2 las condiciones de contorno impuestas se cumplen, continuidad en la posición, en la velocidad y en la aceleración (es decir,  $\theta_1(t), \theta_2(t) \in C^3$ ). Dado que no se han impuesto condiciones sobre la derivada de la aceleración (jerk) se pueden apreciar cambios muy bruscos en esta (es continua pero no derivable), esto implica que esta trayectoria no es apta para ser usada en una máquina real, los cambios bruscos en la aceleración provocarían un gran estrés y desgaste en los motores del robot, acortando su vida útil en el mejor de los casos; es por esto que sería conveniente llevar la suavidad incluso más allá de la tercera derivada (jerk), hasta la cuarta, el jounce.

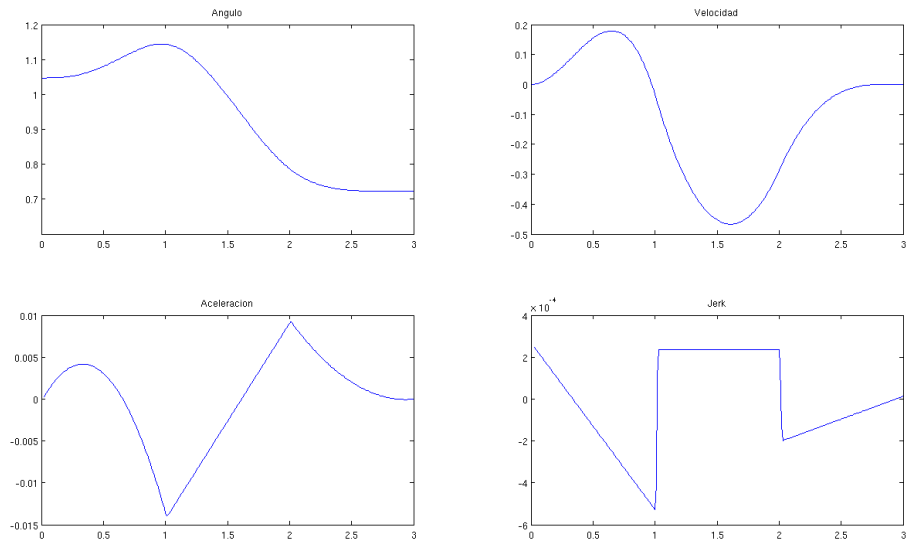


Figura 1.2.1.: Características temporales de  $\theta_1(t)$

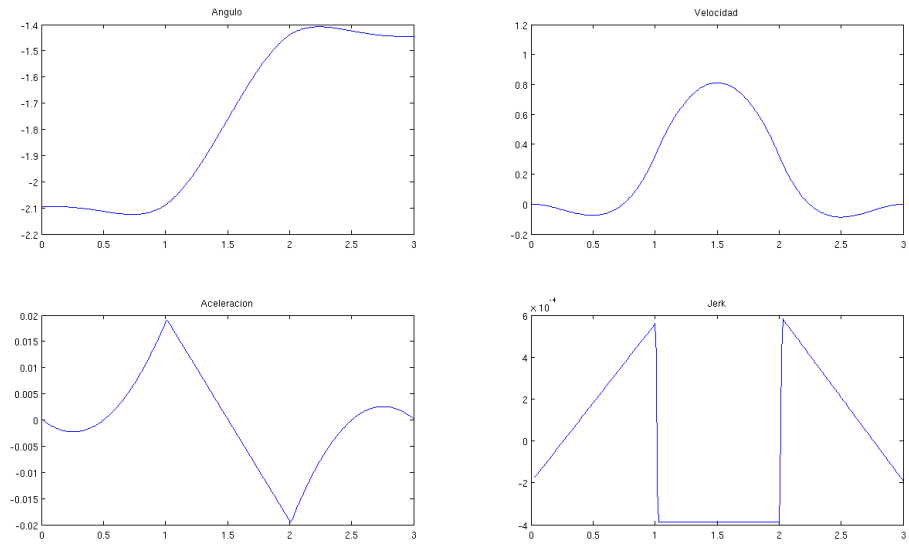


Figura 1.2.2.: Características temporales de  $\theta_2(t)$

## APÉNDICES

## A. CÓDIGO MATLAB UTILIZADO

Listing A.1: Problema cinemático inverso.

```
1 function angulos = pci(L1, L2, p, th1_positivo)
2     x = p(1);
3     y = p(2);
4
5     th2 = acos((x^2 + y^2 - L1^2 - L2^2)/(2*L1*L2));
6     th2 = [th2, -th2];
7
8     th1 = atan2(y * (L1 + L2 * cos(th2)) - x * L2 * sin(th2),...
9             x * (L1 + L2 * cos(th2)) + y * L2 * sin(th2));
10
11     angulos = [th1; th2];
12     if th1_positivo == true
13         indice = find(th1 > 0, 1);
14         if numel(indice) == 1
15             angulos = [th1(indice); th2(indice)];
16         else %no hay positivo, pues el primero mismamente
17             angulos = [th1(1); th2(1)];
18         end
19     end
```

Listing A.2: Problema cinemático directo.

```
1 function p = pcd(L1, L2, th1, th2)
2     px = L1 .* cos(th1) + L2 .* cos(th1 + th2);
3     py = L1 .* sin(th1) + L2 .* sin(th1 + th2);
4     p = [px; py];
```

Listing A.3: Manipulador RR.

```
1 function pxy = robot(L1, th1, p)
2     x=[0 L1 * cos(th1) p(1)];
3     y=[0 L1 * sin(th1) p(2)];
4     pxy = [x; y];
5     plot(x,y)
6     axis([-2 2 -2 2]);
```

Listing A.4: Cálculo de los ángulos de la trayectoria de cada articulación.

```

1 function angulos = calcular_angulos(posiciones, tiempos, t)
2   th1 = calcular_angulo(posiciones(1, :), tiempos, t);
3   th2 = calcular_angulo(posiciones(2, :), tiempos, t);
4   angulos = [th1; th2];
5 end

```

Listing A.5: Cálculo de los valores del ángulo de una articulación.

```

1 function angulo = calcular_angulo(posiciones, tiempos, t)
2   v0 = 0;
3   a0 = 0;
4   vf = 0;
5   af = 0;
6
7   t1 = tiempos(1);
8   t2 = tiempos(2);
9   tn = tiempos(end);
10
11   thini = posiciones(1);
12   thdes = posiciones(2);
13   thase = posiciones(3);
14   thfin = posiciones(4);
15
16   theta0 = thini;
17   theta1 = thdes;
18   theta2 = thase;
19   thetaf = thfin;
20
21   delta1 = theta1 - theta0;
22   delta2 = theta2 - theta1;
23   deltatan = thetaf - theta2;
24
25   Y = [delta1 - (a0*t1^2)/2 - v0*t1;
26        -a0*t1 - v0;
27        -a0;
28        delta2;
29        -af*tn + vf;
30        af;
31        deltatan + (af*tn^2)/2 - vf*tn];
32
33   C = [1 1 0 0 0 0;
34        3/t1 4/t1 -1/t2 0 0 0;
35        6/t1^2 12/t1^2 0 -2/t2^2 0 0;
36        0 0 1 1 0 0;
37        0 0 1/t2 2/t2 3/t2 -3/tn 4/tn;
38        0 0 2/t2^2 6/t2^2 6/tn^2 -12/tn^2;
39        0 0 0 0 1 -1];
40
41   X = C\Y;
42
43   h1 = [X(2), X(1), (a0*t1^2)/2, v0*t1, theta0];
44   h2 = [X(5), X(4), X(3), theta1];
45   hn = [X(7),
46         -4*X(7) + X(6),
47         6*X(7) - 3*X(6) + af*tn^2,
48         -4*X(7) + 3*X(6) - 2*(af*tn^2)/2 + vf*tn,
49         X(7) - X(6) + (af*tn^2)/2 - vf*tn + thetaf
50         ];
51
52   limites_temporales_acumulados = cumsum([0 t1, t2, tn]);
53
54   tramo = find(limites_temporales_acumulados < t, 1, 'last');
55
56   if t >= limites_temporales_acumulados(end)
57     tramo = 3;
58     t_normalizado = 1;
59   elseif t <= 0
60     t_normalizado = 0;
61     tramo = 1;
62   else
63     t_normalizado = (t - limites_temporales_acumulados(tramo))/...
64       (limites_temporales_acumulados(tramo + 1) -...
65        limites_temporales_acumulados(tramo));
66   end
67
68   if tramo == 1
69     angulo = polyval(h1, t_normalizado);
70   elseif tramo == 2
71     angulo = polyval(h2, t_normalizado);
72   elseif tramo == 3

```

```

73     angulo = polyval(hn, t_normalizado);
74     else
75         angulo = -1;
76     end
77 end

```

Listing A.6: Script principal.

```

1  periodo = 0.01;
2  ini = [1; 0];
3  des = [1; 0.1];
4  ase = [1.5; 0.1];
5  ate = [1.5; 0];
6  duraciones = [2; 4; 2];
7
8  angulos_ini = pci(1, 1, ini, true);
9  angulos_des = pci(1, 1, des, true);
10 angulos_ase = pci(1, 1, ase, true);
11 angulos_ate = pci(1, 1, ate, true);
12
13 muestras = 0:periodo:sum(duraciones);
14 angulos_iniciales_cell = repmat(mat2cell([angulos_ini, angulos_des,...
15     angulos_ase, angulos_ate], 2, 4), numel(muestras), 1);
16 duraciones_cell = repmat(num2cell(duraciones, 1), numel(muestras), 1);
17 t_cell = num2cell(muestras');
18 angulos_cell = cellfun(@calcular_angulos, angulos_iniciales_cell,...
19     duraciones_cell, t_cell, 'UniformOutput', false);
20
21 angulos = cell2mat(angulos_cell');
22
23 p = pcd(1, 1, angulos(1,:), angulos(2, :));
24
25 close all;
26
27 for i = 1:numel(muestras)
28     plot(p(1, 1:i), p(2, 1:i));
29     hold(gca, 'on');
30     robot(1, angulos(1, i), p(:, i));
31     hold(gca, 'off');
32     pause(periodo);
33 end
34
35 hold(gca, 'on');
36 puntos_criticos = [ini, des, ase, ate];
37 plot(puntos_criticos(1,:), puntos_criticos(2, :), 'x');
38 hold(gca, 'off');
39
40 pause;
41 close all;
42 subplot(2, 2, 1),
43     plot(muestras, angulos(2, :));
44     title('Angulo');
45 subplot(2, 2, 2),
46     plot(muestras(2:end), diff(angulos(2, :))./diff(muestras));
47     title('Velocidad');
48 subplot(2, 2, 3),
49     plot(muestras(3:end), diff(angulos(2, :), 2)/periodo);
50     title('Aceleracion');
51 subplot(2, 2, 4),
52     plot(muestras(4:end), diff(angulos(2, :), 3)/periodo);
53     title('Jerk');

```