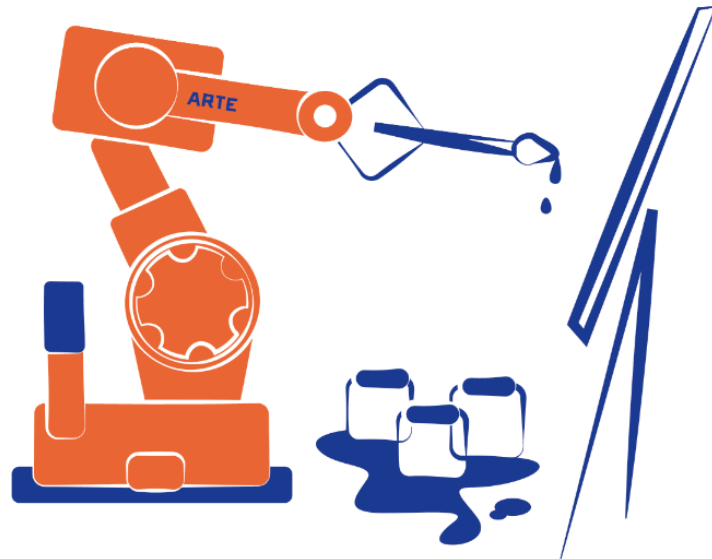


ARTE

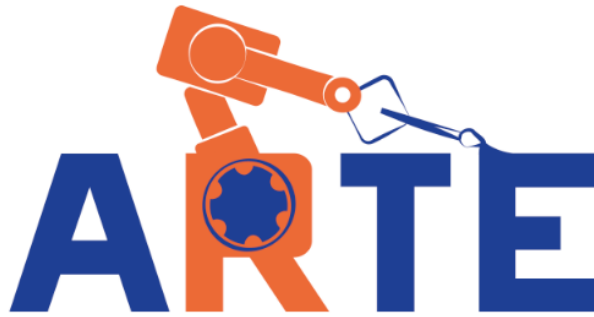


A ROBOTICS TOOLBOX FOR EDUCATION

PRACTICAL SESSION PATH PLANNING

Arturo Gil Aparicio

arturo.gil@umh.es



OBJECTIVES

Después de la realización de la práctica, el estudiante debería ser capaz de:

- Comprender las diferentes estrategias para planificar trayectorias, tanto en el espacio de la tarea como en el espacio articular.
- Implementar en Matlab diferentes estrategias para planificar el movimiento de las articulaciones de un robot.
- Implementar en Matlab diferentes estrategias para posicionar el extremo del robot en una posición/orientación determinadas.
- Considerar la existencia de diferentes soluciones de la cinemática inversa y proponer un algoritmo que pueda planificar una trayectoria continua en el espacio de la tarea.

Index

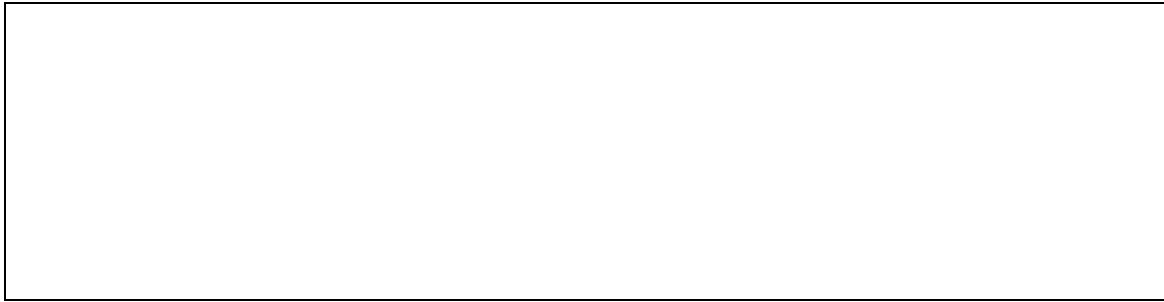
1. First steps
2. Different strategies.
3. Path planning in the joint coordinate space.
4. Sampling a path
5. Following a path your robot to the library
6. Direct and inverse Jacobian
7. Follow a line in space

1 Introducción

Considere que el robot se encuentra en una posición articular determinada \mathbf{q}_0 . Considere, a continuación, que desea ubicar al robot en otra posición articular \mathbf{q}_1 . El robot se deberá mover

Indique, desde su punto de vista qué **datos de partida** necesita Ud. para realizar la planificación.

Seguidamente, evalúe Ud. qué datos de salida serían interesante analizar para evaluar el desempeño de cada uno de los planificadores... p.e. el tiempo total en realizar el movimiento... las velocidades alcanzadas... etc.



Considere que el robot se encuentra en una posición articular determinada \mathbf{q}_0 . Considere, a continuación, que desea ubicar al robot en otra posición articular \mathbf{q}_1 . El robot se deberá mover entre \mathbf{q}_0 y \mathbf{q}_1 esto os debería abrir muchas incógnitas... pues el movimiento que se espera realice el robot debería ser suave. Igualmente, debería ser rápido y preciso. Se espera, también, que el robot evite obstáculos en el entorno. Finalmente, deberíamos contar con alguna función que permitiera que el robot realizara una trayectoria recta en el espacio de la tarea y también cambios de orientación.

Anote aquí qué otras características considera deseables para un módulo de planificación de trayectorias en un robot de tipo serie.

En un lenguaje de programación (RAPID de ABB o KRL de KUKA) son típicas las funciones que permiten realizar una trayectoria recta en el espacio de la tarea interpolando linealmente la posición y orientación del extremo entre dos puntos.

1.1 Definiciones

Debemos distinguir entre:

- **tipos de trayectoria:** Indica cuál es la función que mejor aproxima la trayectoria. Esta trayectoria puede referirse a una trayectoria articular o bien a una trayectoria en el espacio de la tarea. P.e. podemos hablar de una trayectoria de primer orden de la forma $\mathbf{q}_1(t) = \mathbf{k}_1 + \mathbf{k}_2 \cdot t$, donde las constantes \mathbf{k}_1 y \mathbf{k}_2 definen el movimiento de \mathbf{q}_1 en un periodo de tiempo t . Hablaremos de trayectorias de primer orden, de segundo orden... etc.
- **tipos de planificadores:** plantea cómo se calculan las trayectorias anteriores para obtener diferentes resultados con el robot. Hablaremos, en este caso, de planificadores eje a eje, sincronizados... etc.

1.2 Tipos de trayectoria

Trayectorias de primer orden. Responden a la ecuación:

$$q(t) = k_1 + k_2 \cdot t$$

El planificador deberá ser capaz de calcular las constantes para que el movimiento se realice entre dos instantes de tiempo t_1 y t_2 . Se observa, claramente que tendremos una velocidad constante entre ambos puntos articulares.

Trayectorias de segundo orden: En este caso la trayectoria se calcula como:

$$q(t) = k_1 + k_2 \cdot t + k_3 \cdot t^2$$

Vemos que la velocidad es lineal entre ambos puntos y la aceleración es constante a lo largo del movimiento.

1.3 Tipos de planificadores

Conviene en este momento que analicemos soluciones posibles al problema del path planning. Algunas de estas soluciones corresponden a robots fabricados en el pasado y que podríamos considerar poco avanzados tecnológicamente.

1.3.1 Movimiento eje a eje

Sencillo. Solamente una articulación se mueve en cada instante de tiempo. La trayectoria del extremo del robot no puede seguir una línea recta. El tiempo total que dura la trayectoria corresponde a la suma de los movimientos de todas las articulaciones que se realizan de forma consecutiva.

1.3.2 Movimiento simultáneo de ejes

Sencillo. Varias articulaciones inician su movimiento al mismo tiempo. Todas las articulaciones se mueven a su velocidad máxima. El movimiento de las articulaciones no finaliza en el mismo instante de tiempo.

1.3.3 Movimiento coordinado isócrono de ejes

Las articulaciones se mueven al tiempo. El movimiento de las articulaciones comienza y finaliza en el mismo instante de tiempo. Generalmente, alguna de las articulaciones funcionará en su velocidad máxima, según el número de radianes que deba moverse.

1.3.4 Movimiento en el espacio de la tarea

Plantea un nivel de abstracción mayor a los comentados anteriormente. Primero se define una trayectoria del extremo del robot. Esta puede consistir, por ejemplo, en:

- Una trayectoria recta a velocidad constante y orientación constante.
- Una trayectoria recta con un perfil trapezoidal de velocidades.

- Una reorientación del extremo a posición constante.
- Una trayectoria circular entre 3 puntos.
- Una trayectoria circular con reorientación.

2 Ejercicios

Realice los ejercicios que se plantean en
arte/exercises/path_planning/interpolators.m

Ejercicio 1: Programe un planificador de primer orden.

Planifique una trayectoria articular desde dos valores con un planificador de primer orden. El movimiento debe realizarse a una velocidad constante de 2 rad/s. Muévase de $q(0)=0$ a $q(T)=1.8$ rad.

- Complete la función `first_order` que devuelva los datos necesarios.
- Dibuje la trayectoria en posición con variaciones de 0.001 segundos. Plotee también la velocidad y la aceleración en cada tiempo.
- Presente sus conclusiones. Desde su punto de vista, ¿qué aspectos deberían mejorarse? Observe que la aceleración toma valores infinitos en el inicio y final del movimiento y que esto podría dar lugar a la aplicación de pares innecesariamente altos en las articulaciones.

El ejercicio 1 se anota aquí comentado su solución.

****1**** --> dada la velocidad máxima se calcula el vector de tiempo en base a la trayectoria a realizar.

****2**** --> en este punto, se calculan los coeficientes del de la ecuación de la trayectoria. Tenga Ud. en cuenta, que debe clumplirse.

$$q(t=0) = k_1 = 0 \text{ rad}$$

$$q(t=T_2) = k_1 + k_2 \cdot T_2 = 1.8 \text{ rad}$$

La solución de este problema se obtiene como $k = \text{inv}(A) \cdot b'$;

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%   Exercise 1:
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function exercisel()
% Go from q1=0 to q2=1.8 rad
q=[0 1.8];
% At a constant speed.
qd=2; %rad/s
% DEFINE t = [T1 T2]
T1 = 0;
T2 = (q(2)-q(1))/qd;

% (**1**)

t=[T1 T2];
%the minimum difference in seconds between times to compute q(t)
functions
delta_t=0.001;
```

```

% FIRST ORDER PLANNER FROM q1 to q2
figure, xlabel('t (s)'), ylabel('q (rad), q_d (rad/s)'), title('FIRST
ORDER PLANNER. EXERCISE 1'), hold on
[q_t, qd_t, time, k]=first_order([q(1) q(2)], [t(1) t(2)], delta_t);
plot(time, q_t, 'r'), plot(time, qd_t, 'g')

legend('Position (rad)', 'Speed (rad/s)')
%WRITE EQUATION
disp('The equation computed can be written as:')
fprintf('\n q(t)=k1+k2*t=%.3f+%.3f*t', k(1), k(2))

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% First order interpolator.
%   Given the start and end joint positions and times compute a first
%   order polinomial of the form:
%   q(t) = k(1) + k(2)*t
%
%   Inputs:
%   q: a joint vector of two inputs q = [q1 q2]
%   t: a time vector with t = [t1 t2]
%   delta_t: the sample time between computed q(t)'s
%   Returns:
%
%   q: the values of q(t) as a function of the time vector used.
%   qd: the speed qd(t) as a function of the time vector.
%   t: the time vector used.
%   k: the polynomial coefficients k that allow to compute q(t)
as:
%           q(t) = k(1) + k(2)*t
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [q_t, qd_t, time, k]=first_order(q, t, delta_t)
% define the time matrix

% (**2**)

A = [1 t(1);
     1 t(2)];
b = [q(1) q(2)];
% compute the coefficients of k for a first order planner
k = inv(A)*b';

% define the time vector using delta_time
time=t(1):delta_t:t(2);
% the first order equation
q_t=k(1) + k(2)*time;
% return a constant speed
qd_t=k(2)*ones(1,length(time));

```

Dado que conoce q_t , use la función `animate` con un robot y observe su movimiento.

Ejercicio 2: Derive un planificador de segundo orden.

Escriba un script de Matlab para el cálculo de una trayectoria de la forma:

$$q(t) = k_1 + k_2*t + k_3*t^2$$

Para ello, escriba una función que tenga la forma siguiente:

```
[q_t, qd_t, qdd_t, t, k]=second_order(q, qd, t, delta_t)
```

Con entradas:

q: un vector con los valores iniciales y finales de la trayectoria para una articulación. $q = [q_1 \ q_2]$

qd: la velocidad inicial en t_1 .

t: el vector del tiempo inicial y final con $t = [t_1 \ t_2]$

delta_t: tiempo entre muestras para construir $q(t)$

Salidas:

q: valores de $q(t)$ como función del vector temporal

qd: velocidad como función del vector de tiempo usado.

t: el vector de tiempo usado.

k: el vector de coeficientes que permite calcular la trayectoria como:

$$q(t) = k(1) + k(2)*t + k(3)*t^2$$

Finalmente presente sus conclusiones. ¿Qué aspectos se deberían mejorar en este planificador, desde su punto de vista?

Ejercicio 3: Utilice varios interpoladores lineales en la trayectoria.

Realice las siguientes cuestiones:

a) Use un planificador de primer orden para interpolar una trayectoria en una sola articulación a lo largo del tiempo. Considere las siguientes coordenadas en el espacio de la articulación y el tiempo.

$q_1 = 0.2 \text{ rad. } t_1 = 0 \text{ s}$

$q_2 = 0.5 \text{ rad. } t_2 = 6 \text{ s}$

$q_3 = 0.8 \text{ rad. } t_3 = 8 \text{ s}$

$q_4 = 1.5 \text{ rad. } t_4 = 10 \text{ s}$

b) Dibuje en Matlab la trayectoria total en velocidad y posición. c) Indique sus conclusiones principales, desde su punto de vista.

Ejercicio 4: Realice un perfil de aceleración trapezoidal con un interpolador de 4 orden.

Los valores conocidos para este problema son:

- q_{ini} , q_{final} : los valores iniciales y finales de la trayectoria (rad).
- a_{max} : máxima aceleración angular permitida. Este es un parámetro que se supone conocido del robot.
- w_{max} : máxima velocidad angular permitida.

Realice las siguientes cuestiones:

- El primer sector de aceleración trapezoidal se proporciona, como ejemplo, al estudiante.
- Complete el segundo y tercer sectores del perfil trapezoidal (primer sector: aceleración constante; segundo sector: velocidad constante; tercer sector: deceleración constante).
- Dibuje en Matlab los resultados.

Ejercicio 5: Realice una animación de un robot de 3 GDL con la trayectoria calculada en el ejercicio 4.

Repita los resultados obtenidos en el ejercicio 4 para cada una de las 3 articulaciones y utilice la función `animate` para simular el movimiento del robot `3dofplanarrobot`

Ejercicio 6: Seguir una trayectoria en el espacio

Seguir una trayectoria recta en el espacio es una tarea común y frecuente de un robot industrial en cualquier aplicación. Un operador humano encuentra, generalmente, fácil, la realización de un movimiento por parte de un robot que siga una trayectoria recta entre dos puntos en un espacio cartesiano que conecte dos posiciones/orientaciones del robot. En general, seguir una trayectoria recta en el espacio de la tarea no será óptimo en términos de velocidad o eficiencia pero, aún así, es posible encontrar una gran cantidad de aplicaciones en las que se precisa este tipo de aplicaciones, por ejemplo:

- a) Soldadura TIG de dos planchas de metal a lo largo de una línea.
- b) Soldadura por puntos a lo largo de una línea. Point welding es típico en la industria de fabricación de vehículos automóviles.

Ejercicio 6:

Implemente una función como la siguiente:

function q=followline_myrobot(robot, p1, p2, R, options)

Que permita seguir una línea por el extremo del robot en el espacio cartesiano. La función debe seguir un conjunto de puntos sobre la línea que conecta los puntos p_1 y p_2 en el espacio cartesiano. La orientación debería mantenerse constante al valor definido por la matriz R . En cada uno de los puntos que conectan esta recta es necesario obtener la solución de la cinemática inversa. El parámetro `options` debe usarse para poder elegir entre las soluciones existentes de la cinemática inversa. Considere que el movimiento final del robot debe ser continuo, en consecuencia, no se admite que existan saltos importantes entre diferentes soluciones a la cinemática inversa. Una solución típica consiste en elegir una de las soluciones de la cinemática inversa para el punto p_1 y a continuación, elegir las soluciones más cercanas en el espacio euclídeo de las articulaciones..

Finalmente, Ud. debería:

- a) Animar la trayectoria del robot.
- b) Representar las trayectorias articulares.
- c) Plotear las velocidades articulares asumiendo que el movimiento entre p_1 y p_2 se realiza en 1 segundo.
- d) Enumera los posibles errores que pueden aparecer durante la consecución de la tarea como, por ejemplo:
 - Puntos fuera del espacio de trabajo del robot.
 - Puntos singulares donde $\det(J)=0$. En estos puntos singulares pueden aparecer velocidades infinitas para intentar conseguir velocidades finitas del extremo.

- Valores articulares fuera de rango.