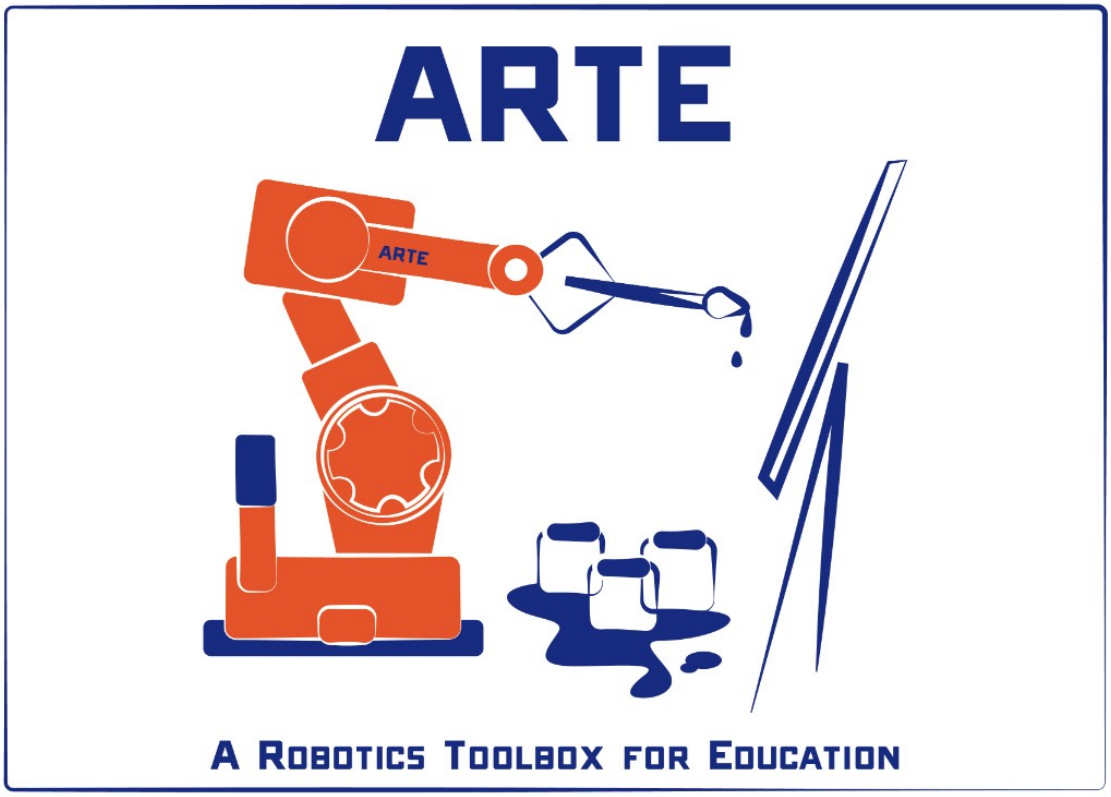
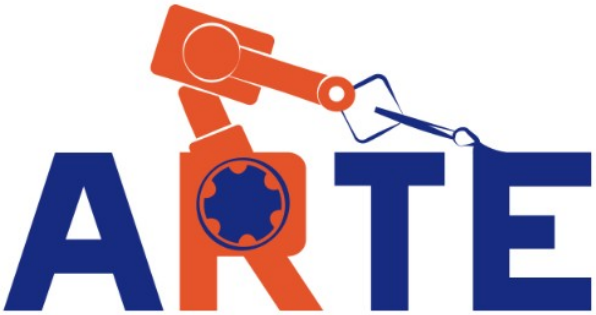


Sesión 2 C: Conceptos avanzados de Path Planning. Seguimiento de rectas en el espacio de trabajo.

Arturo Gil Aparicio

arturo.gil@umh.es



OBJETIVOS

Objetivos generales: Después de esta sesión práctica, el estudiante debería ser capaz de:

- Utilizar la Jacobiana del manipulador para seguir trayectorias rectas en el espacio.
- Entender el error cometido en el seguimiento de estas trayectorias y plantear formas de reducirlo.
- Aplicar las trayectorias planificadas en el simulador Coppelia Sim.

Índice:

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, por tanto, desde unos valores iniciales de los ángulos de las articulaciones hasta unos valores finales.

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 de las articulaciones debería ser suave… pero… ¿qué significa 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.

Por otra parte, en un lenguaje de programación (RAPID de ABB o KRL de KUKA) son típicas dos tipos de funciones:

- funciones que permiten llevar al robot entre dos posiciones articulares.

- funciones que permiten realizar una trayectoria recta en el espacio de la tarea interpolando linealmente la posición y orientación del extremo entre dos puntos.

- Otro tipo de trayectorias: p.e. trayectorias circulares.

1.1 Definiciones

Debemos distinguir entre:

a) 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 $q_1(t) = k_1 + k_2 \cdot t$, donde las constantes k1 y k2 definen el movimiento de q1 en un periodo de tiempo t. Hablaremos de trayectorias de primer orden, de segundo orden... etc.

b) 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) = k1 + k2*t
```

El planificador deberá ser capaz de calcular las constantes para que el movimiento se realice entre dos instantes de tiempo t1 y t2. 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) = k1 + k2*t + k3*t^2
```

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

Podemos plantear otros planificadores de mayor orden.

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.

3 Instalación de Coppelia Sim

Se propone instalar Coppelia Sim en tu PC.

1) Descargar Coppelia sim para tu plataforma (Windows/Ubuntu/Mac). Puedes descargar la versión EDU de

https://www.coppeliarobotics.com/downloads

2) Descargar la última versión de ARTE.

En ARTE, encontraréis:

- ARTE/coppeliaSim → práctica de hoy.
- ARTE/coppeliaSim/coppelia_api → deberás guardar aquí los ficheros de tu plataforma
- ARTE/coppeliaSim/artecoppeliabind → helper functions para relacionar ARTE con Coppelia.

- En windows/Mac: encuentra remoteApi.dll, remoteApi.dylib o remoteApi.so (depending on your target platform). Estos ficheros deben estar en la carpeta de instalación en programming/remoteApiBindings/matlab

Coppelia Sim tiene una serie de características que la separa de ARTE:

- Principalmente, ARTE no plantea la simulación a tiempo real del sistema. Los gráficos en Matlab a tiempo real son una cuestión que depende de la aceleración gráfica del sistema y son difícilmente generalizables.
- CoppeliaSim plantea una simulación a tiempo real. En concreto:
- Permite enviar comandos de control en velocidad, posición y par para que los siga el robot.
- La simulación se hace considerando un paso fijo dt=50 ms.
- Además, tiene en cuenta auto-colisiones, colisiones con piezas del entorno, colisiones con otros robots… etc.

La librería ARTE se conecta a través de una API con Coppelia. Existen APIS para C/C++, python, Matlab y Octave.

4 Pruebas iniciales con Coppelia Sim

Ejecute Coppelia Sim. Puede explorar sus directorios de objetos para descubrir robots de tipo serie, paralelo y sensores, entre otros. El profesor dará en este momento una introducción sobre este simulador. En concreto:

- Objetos que se pueden importar.
- Objetos básicos que se pueden añadir. Posición, orientación y propiedades.
- Articulaciones qi, sistema de control e interfaz vía API.

Existen algunos entornos de Coppelia ya guardados en ARTE especialmente preparados para ser manejados desde ARTE. En concreto, es necesario tener unas convenciones:

- Las articulaciones de los robots se denominan qi. Por ejemplo, q1, q2, q3, … q6. Eso permite, a partir de una cadena de texto, obtener una referencia para controlar una articulación en posición/orientación.

- Debe existir unas funciones denominadas close_gripper y open_gripper para el manejo de la pinza del robot de forma abstracta.

Cargue el entorno que se encuentra en

ARTE/coppeliaSim/scenes/irb140_abb.ttt

5 Seguimiento de trayectorias en el espacio de trabajo

Se propone al estudiante trabjar con el fichero

ARTE/coppeliaSim/irb140_abb_follow_line.m

Se propone comandar al robot usando diferentes tipos de trayectoria.

- AbsJPath-→ se especifica una Q en términos absolutos y el robot debe seguir una trayectoria en el espacio de las articulaciones hasta esa Q.
- MoveLPath → Se especifica una T y el robot debe seguir una trayectoria recta hasta esa T.

En nuestra aplicación se propone al estudiante que realice las acciones necesarias para que el robot coja una pieza del suelo (cubo, esfera…, etc) y la deposite sobre una plataforma cilíndrica).

En esta práctica se proporcionan algunas funciones ya realizadas, en concreto.

AbsJPath: Permite especificar posición articular y velocidad articular siguiente. Se indica el porcentaje de velocidad deseado sobre el máximo permitido por el robot.

Actividades:

- A) MoveLPath: Esta función **la deberá mejorar el estudiante durante la práctica.**
- B) MoveQPath: Esta función **la deberá crear el estudiante durante la práctica.**
- C) Cree una tarea donde el robot coja una o varias piezas y las coloque donde el estudiante desee.

Note el estudiante que, como es común en todas las instrucciones de movimiento en Robótica serie, se especifica siempre:

- La siguientes coordenadas articulares o velocidades articulares. El planificador realiza esta acción considerando conocidas las posiciones/velocidades articulares en el instante anterior.

Actividad A:

La ejecutar el script irb140_abb_follow_line.m nos daremos cuenta de que la trayectoria no se puede seguir de forma fiel. En concreto, en cada paso de nuestra planificación, cuando hacemos

qd = inv(J)*Vref

estamos cometiendo un pequeño error que no está acotado. De esta manera, el error de seguimiento (error_line) vemos que crece en cada iteración. Proponga una solución a este problema. En el script de seguimiento de línea cuenta ya con la función:

```
[delta_end, error_line, error_line_vector] = find_errors(start_point,
end_point, p);
```

Dicha función proporciona un vector unitario en dirección con el mínimo error del extremo y la recta. Plantee, también, formas de corregir el error de orientación. Tenga en cuenta que se proporciona la variable `error_line_vector`, que define un vector perpendicular a la recta y que pasa por el extremo del robot. Plantee una solución que intente minimizar este error.

Actividad B:

Cree una función que se denomine MoveQPath. Esta función debe tener el siguiente prototipo.

```
[qt, qdt] = MoveLPath(robot, T, speed_percent);
```

Esta función debe hacer una trayectoria isócrona en el espacio de las articulaciones. Sin embargo, como partida, se da una matriz de posición y orientación. El parámetro speed_percent indica un porcentaje respecto de la velocidad máxima de cada articulación. Puede plantear, por ejemplo, una planificación lineal de todas las articulaciones. El estudiante debe pensar cómo especificar una de las soluciones de la cinemática inversa.