

SISTEMA AMBIDEXTRO DE MANIPULACIÓN ROBÓTICA

- Para hacer este proyecto lo más flexible posible hemos realizado un script llamado [trayectorias.m](#) que nos genera la trayectoria entre el punto inicial dado y el punto final deseado, siguiendo el camino más óptimo *
- Como para realizar la interpolación entre el punto inicial y el punto final es necesario tres coordenadas de posición y tres de orientación, hemos realizado otro fichero llamado [calcula_puntos.m](#) para su cálculo

- Continuando con la función [trayectorias.m](#), derivamos la posición para obtener la velocidad. Hemos utilizado:

- Velocidad, posición y rotación
- Moore-Penrose
- Velocidades articulares

$$\begin{bmatrix} \dot{q}_1 \\ \vdots \\ \vdots \\ \dot{q}_n \end{bmatrix} = \mathbf{J}^{-1} \begin{bmatrix} \dot{x} \\ \vdots \\ \vdots \\ \dot{\gamma} \end{bmatrix}$$

$$\mathbf{W}(t) = \begin{pmatrix} 0 & -\omega_z(t) & \omega_y(t) \\ \omega_z(t) & 0 & -\omega_x(t) \\ -\omega_y(t) & \omega_x(t) & 0 \end{pmatrix}$$

- Sacamos \mathbf{r} (matriz de rotación) y con [obtencion_w.m](#) usamos esas \mathbf{r} para conocer las velocidades angulares, \mathbf{w} . Este método sirve para cualquier tipo de trayectoria entre posición de inicio y fin. Por lo tanto, nos resolverá cualquier trayectoria, por arbitraria que sea, tanto si es recta como si es curvilínea

* Se ha usado *teach* (función perteneciente a la librería ARTE) para obtener una guía de qué puntos son válidos y cuáles son singulares

- Debemos editar los parámetros iniciales del robot 2 cambiando en *parameters.m* los valores de la tabla DH y poniendo en el primer valor de theta = $q(1)+\pi$. Cambiamos también la posición de la base, que quedaría: Coordenadas base robot 1: (0, 0, 0) y Coordenadas base robot 2: (0, -1, 0)
- Una vez configurada la posición y orientación inicial y final, para que cada brazo empiece a un lado de la caja y ambos terminen en la posición y orientación final deseada para la caja, a continuación, generamos otro fichero llamado *obtencion_q.m*
- En el script *demo.m* la función utiliza los parámetros robot1, robot2 y caja, ya que esta función es la encargada de cuadrar los diseños a la misma vez en el grid
- Con *drawrobot3d* dibujamos ambos robots (en el interior de un bucle)
- Se ha editado el fichero *drawrobot3d* de la librería ARTE para poder representar ambos robots simultáneamente junto con la caja *fruit_box* y *la caja*:

```
% h=figure(configuration.figure.robot);  
h=0;
```