

---

## Generación de trayectorias

---

Introducción.....	156
Instrucciones MATLAB relacionadas con la generación de trayectorias.....	156
Ejemplos.....	165
Referencias.....	165

## Introducción

Para tratar la generación de trayectorias se dispone de una serie de funciones MATLAB. Se cubren tanto la generación de trayectorias en el espacio cartesiano como en el espacio de las variables articulares.

Para la generación de trayectorias en el espacio de las variables articulares se dispone de funciones de interpolación cúbicas, quinticas y de funciones lineales con enlaces parabólicos. En todos estos casos, las ecuaciones empleadas para la implementación de las funciones han sido extraídas del Capítulo 10 de Ollero [2]. También se dispone de una función para la generación de trayectorias articulares basada en polinomios de orden 7 (Corke [1]).

En cuanto a la generación de trayectorias en el espacio cartesiano, se cuenta con una serie de funciones extraídas de Corke [1], cuyo uso ha sido ilustrado mediante el uso de la función `frame` descrita en el Capítulo 2 del presente manual.

Algunas de las funciones presentadas en este capítulo se emplean en el Capítulo 10 de Ollero [2].

## Instrucciones MATLAB relacionadas con la generación de trayectorias

En la siguiente tabla se resumen las instrucciones que se pueden emplear en relación con la generación de trayectorias, tanto articulares como cartesianas:

Tabla 7.1: Instrucciones relacionadas con la generación de trayectorias.

	Propósito
<code>jtraj</code>	Calcular una trayectoria articular entre dos posiciones articulares mediante el uso de polinomios de orden siete
<code>parab</code>	Calcular las trayectorias articulares resultantes de emplear funciones lineales con enlace parabólico
<code>pol3</code>	Calcular las trayectorias articulares resultantes de emplear polinomios cúbicos
<code>pol5</code>	Calcular las trayectorias articulares resultantes de emplear polinomios quinticos
<code>trinterp</code>	Calcular la interpolación entre dos transformaciones homogéneas

---

## jtraj

---

### Propósito

Calcular una trayectoria articular entre dos posiciones articulares mediante el uso de polinomios de orden siete.

### Sintaxis

```
[q qd qdd]=jtraj(q0,q1,n)
[q qd qdd]=jtraj(q0,q1,n,qd0,qd1)
[q qd qdd]=jtraj(q0,q1,t)
[q qd qdd]=jtraj(q0,q1,t,qd0,qd1)
```

### Descripción

Jtraj devuelve una trayectoria de posiciones articulares ( $q$ ) que enlaza la posición articular  $q0$  con la  $q1$ . El número de puntos que se calcula viene dado por el parámetro  $n$ , o bien por la longitud del vector  $t$ . Se utiliza un polinomio de orden siete con condiciones iniciales y finales nulas por defecto para la velocidad y la aceleración.

Se pueden utilizar los parámetros  $qd0$  y  $qd1$  para especificar velocidades articulares iniciales y finales no nulas.

Las trayectorias articulares devueltas por la función son matrices, con una fila por cada articulación y una columna por cada instante de tiempo. Se calculan tanto las posiciones como las velocidades y las aceleraciones articulares.

### Ver también

parab, pol3, pol5

### Referencias

Corke, P.I., *A robotics toolbox for MATLAB*, IEEE Robotics and Automation Magazine, Vol.3, núm. 1, pp. 24-32, 1996.

---

## parab

---

### Propósito

Calcular las trayectorias articulares resultantes de emplear funciones lineales con enlace parabólico.

### Sintaxis

`[qcoef,time,q_t,qd_t,qdd_t,intervs]=parab(q,ac,t0,tij,step)`

### Descripción

Esta función calcula las trayectorias articulares resultantes de emplear funciones lineales con enlaces parabólicos para la interpolación. Los parámetros de entrada de la función son los siguientes:

- **q:** matriz con tantas filas como articulaciones, que contiene en cada columna las posiciones articulares deseadas para el punto inicial, los puntos de paso y el punto final.
- **ac:** matriz con tantas filas como articulaciones, que contiene en cada columna los módulos de las aceleraciones articulares deseados para el punto inicial, los puntos de paso y el punto final.
- **t0:** parámetro que contiene el instante inicial.
- **tij:** vector que contiene la duración de los intervalos entre puntos.
- **step:** paso con el que se calculan la posición, velocidad y aceleración resultantes.

Esta función devuelve:

- **qcoef:** matriz de tres dimensiones, que contiene en cada fila los coeficientes de los polinomios de orden dos, que son solución para cada una de las variables articulares. La primera columna se corresponde con la potencia cuadrática y la última con el término independiente. Para los tramos lineales, el término cuadrático es nulo. El tercer índice de la matriz hace referencia a la articulación a la que le corresponden los coeficientes.
- **time:** vector tiempo con los instantes para los que se calculan las matrices **q\_t**, **qd\_t** y **qdd\_t** que contienen los valores de posición, velocidad y aceleración articular resultantes.
- **q\_t:** matriz que contiene en cada fila los valores de posición correspondientes a cada articulación.

- `qd_t`: matriz que contiene en cada fila los valores de velocidad correspondientes a cada articulación.
- `qdd_t`: matriz que contiene en cada fila los valores de aceleración correspondientes a cada articulación.
- `intervs`: matriz que contiene en cada fila las duraciones de los tramos parabólicos y lineales para cada articulación.

## Ejemplos

Ver Ejemplos 10.5 y 10.6 de Ollero [2].

## Ver también

`jtraj`, `pol3`, `pol5`

## Referencias

Ollero, A., *Robótica: Manipuladores y robots móviles*, Marcombo-Boixareu editores, 2001.

# pol3

## Propósito

Calcular las trayectorias articulares resultantes de emplear polinomios cúbicos.

## Sintaxis

`[qcoef,time,q_t,qd_t,qdd_t]=pol3(q,qd,t0,tij,step)`

## Descripción

Esta función calcula las trayectorias articulares resultantes de emplear polinomios cúbicos para la interpolación. Los parámetros de entrada de la función son los siguientes:

- **q:** matriz con tantas filas como articulaciones, que contiene en cada columna las posiciones articulares deseadas para el punto inicial, los puntos de paso y el punto final.
- **qd:** matriz con tantas filas como articulaciones, que contiene en cada columna las velocidades articulares deseadas para el punto inicial, los puntos de paso y el punto final.
- **t0:** parámetro que contiene el instante inicial.
- **tij:** vector que contiene la duración de los intervalos entre puntos.
- **step:** paso con el que se calculan la posición, velocidad y aceleración resultantes.

Esta función devuelve:

- **qcoef:** matriz de tres dimensiones, que contiene en cada fila los coeficientes del polinomio cúbico solución para cada una de las variables articulares. La primera columna se corresponde con la potencia cúbica y la última con el término independiente. El tercer índice de la matriz hace referencia a la articulación a la que le corresponden los coeficientes.
- **time:** vector tiempo con los instantes para los que se calculan las matrices **q\_t**, **qd\_t** y **qdd\_t** que contienen los valores de posición, velocidad y aceleración articular resultantes.
- **q\_t:** matriz que contiene en cada fila los valores de posición correspondientes a cada articulación.

- `qd_t`: matriz que contiene en cada fila los valores de velocidad correspondientes a cada articulación.
- `qdd_t`: matriz que contiene en cada fila los valores de aceleración correspondientes a cada articulación.

## Ejemplos

Ver Ejemplos 10.1 y 10.2 de Ollero [2].

## Ver también

`jtraj`, `parab`, `pol5`

## Referencias

Ollero, A., *Robótica: Manipuladores y robots móviles*, Marcombo-Boixareu editores, 2001.

---

## pol5

---

### Propósito

Calcular las trayectorias articulares resultantes de emplear polinomios quinticos.

### Sintaxis

`[qcoef,time,q_t,qd_t,qdd_t]=pol5(q,qd,qdd,t0,tij,step)`

### Descripción

Esta función calcula las trayectorias articulares resultantes de emplear polinomios quinticos para la interpolación. Los parámetros de entrada de la función son los siguientes:

- **q:** matriz con tantas filas como articulaciones, que contiene en cada columna las posiciones articulares deseadas para el punto inicial, los puntos de paso y el punto final.
- **qd:** matriz con tantas filas como articulaciones, que contiene en cada columna las velocidades articulares deseadas para el punto inicial, los puntos de paso y el punto final.
- **qdd:** matriz con tantas filas como articulaciones, que contiene en cada columna las aceleraciones articulares deseadas para el punto inicial, los puntos de paso y el punto final.
- **t0:** parámetro que contiene el instante inicial.
- **tij:** vector que contiene la duración de los intervalos entre puntos.
- **step:** paso con el que se calculan la posición, velocidad y aceleración resultantes.

Esta función devuelve:

- **qcoef:** matriz de tres dimensiones, que contiene en cada fila los coeficientes del polinomio quintico solución para cada una de las variables articulares. La primera columna se corresponde con la potencia cúbica y la última con el término independiente. El tercer índice de la matriz hace referencia a la articulación a la que le corresponden los coeficientes.
- **time:** vector tiempo con los instantes para los que se calculan las matrices **q\_t**, **qd\_t** y **qdd\_t** que contienen los valores de posición, velocidad y aceleración articular resultantes.

- `q_t`: matriz que contiene en cada fila los valores de posición correspondientes a cada articulación.
- `qd_t`: matriz que contiene en cada fila los valores de velocidad correspondientes a cada articulación.
- `qdd_t`: matriz que contiene en cada fila los valores de aceleración correspondientes a cada articulación.

## Ejemplos

Ver Ejemplos 10.3 y 10.4 de Ollero [2].

## Ver también

`jtraj`, `parab`, `pol3`

## Referencias

Ollero, A., *Robótica: Manipuladores y robots móviles*, Marcombo-Boixareu editores, 2001.

## trinterp

---

### Propósito

Calcular la interpolación entre dos transformaciones homogéneas.

### Sintaxis

`T=trinterp(T0,T1,r)`

`T=trinterp(T0,dp,r)`

### Descripción

Esta función, en su primera sintaxis, interpola entre las dos transformaciones homogéneas `T0` y `T1` a medida que `r` varía entre 0 y 1. Se suele usar generalmente para generar trayectorias cartesianas.

La segunda sintaxis emplea el vector `dp` devuelto por la función `drivepar` (que representa la ‘diferencia’ entre `T0` y `T1`) y puede ser más eficiente cuando se calculen muchos puntos interpolados entre las mismas transformaciones.

### Ver también

`ctray`, `drivepar`

### Referencias

Corke, P.I., *A robotics toolbox for MATLAB*, IEEE Robotics and Automation Magazine, Vol.3, núm. 1, pp. 24-32, 1996.

Paul, R.P., *Robot manipulators: Mathematics, Programming, and Control*. Cambridge, Massachusetts: MIT Press, 1981.

## Ejemplos

*Ejemplo H. 7.1* (archivo `ejh71.m`): Ver Ejemplo 10.1 en Ollero [2].

*Ejemplo H. 7.2* (archivo `ejh72.m`): Ver Ejemplo 10.2 en Ollero [2].

*Ejemplo H. 7.3* (archivo `ejh73.m`): Ver Ejemplo 10.3 en Ollero [2].

*Ejemplo H. 7.4* (archivo `ejh74.m`): Ver Ejemplo 10.4 en Ollero [2].

*Ejemplo H. 7.5* (archivo `ejh75.m`): Ver Ejemplo 10.5 en Ollero [2].

*Ejemplo H. 7.6* (archivo `ejh76.m`): Ver Ejemplo 10.6 en Ollero [2].

## Referencias

- [1] Corke, P.I., *A robotics toolbox for MATLAB*, IEEE Robotics and Automation Magazine, Vol.3, núm. 1, pp. 24-32, 1996.
- [2] Ollero, A., *Robótica: Manipuladores y robots móviles*, Marcombo-Boixareu editores, 2001.

