



UNIVERSIDAD SIMÓN BOLÍVAR  
DEPARTAMENTO DE ELECTRÓNICA Y CIRCUITOS  
FUNDAMENTOS DE MECATRÓNICA (EC5811)

# Proyecto

Diseño general de un manipulador universal

**Estudiantes:**

Jesús Colmenares Carnet: 14-11384

Jesús Guillén Carnet: 14-10460

**Profesor:**

Juan Grieco

Sartenejas, junio de 2019

## **Contenido**

- *Introducción*
- *Objetivos*
- *Estructura física del manipulador*
- *Cinemática directa*
- *Cinemática inversa*
- *Velocidad cinemática*
- *Dinámica del manipulador*
- *Planificador de trayectorias*
- *Pruebas generales*
- *Conclusiones*
- *Referencias*
- *Anexo: ejecución del programa*

## **Introducción**

En las últimas décadas, la robótica ha revolucionado drásticamente los procesos de fabricación hasta límites inimaginables. Esto ha permitido aumentar la producción, mejorar el nivel de calidad y librar a las personas de efectuar labores pesadas y riesgosas.

El crecimiento de esta área es tan grande, especialmente en los sectores automotriz, electrónico y metalúrgico, que se estima que para el 2020 estén operativos más de 3 millones de robots industriales. En este sentido los brazos robóticos juegan un papel fundamental ya que son de los robots industriales más utilizados. En este grupo se destacan los manipuladores universales, que son brazos con 6 grados de libertad lo que significa que pueden manipular un objeto en cualquier lugar (dentro de su espacio de trabajo) y en cualquier orientación.

En el presente trabajo, valiéndose del software MATLAB y específicamente del Robotics Toolbox de Peter Corke, se procederá al diseño de un manipulador universal inspirado en el robot PUMA560, contando con 6 articulaciones rotacionales.

Se efectuará su análisis a nivel de cinemática (directa e inversa), velocidad cinemática, así como también su dinámica básica. Estos estudios se efectuarán sobre puntos particulares y en trayectorias.

## **Objetivos**

### **General:**

- Realizar un diseño general y de alto nivel de un manipulador universal.

### **Específicos:**

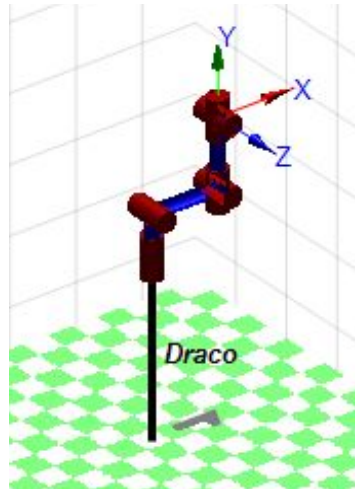
- Definir la estructura física del manipulador: sus eslabones, articulaciones y espacio de trabajo.
- Estudiar la cinemática directa.
- Estudiar la cinemática inversa.
- Construir un planificador de trayectorias.
- Estudiar la velocidad cinemática.
- Encontrar un modelo dinámico completo.

## **Estructura física del manipulador**

La estructura general del manipulador debe cumplir con el requerimiento principal del proyecto: debe ser universal. Por ello, se tiene que el mismo debe tener seis grados de libertad. Para garantizar esto, se plantean seis grados de movilidad, de manera que no haya redundancia en las articulaciones. De este modo, se puede colocar un objeto en cualquier posición y orientación dentro del espacio de trabajo del robot. A continuación se brindará una descripción general y de alto nivel de la estructura física del manipulador, definiendo cada articulación, su tipo, rango de movilidad y espacio de trabajo. Los detalles más específicos de implementación como diseño particular de cada pieza están fuera del alcance de este proyecto.

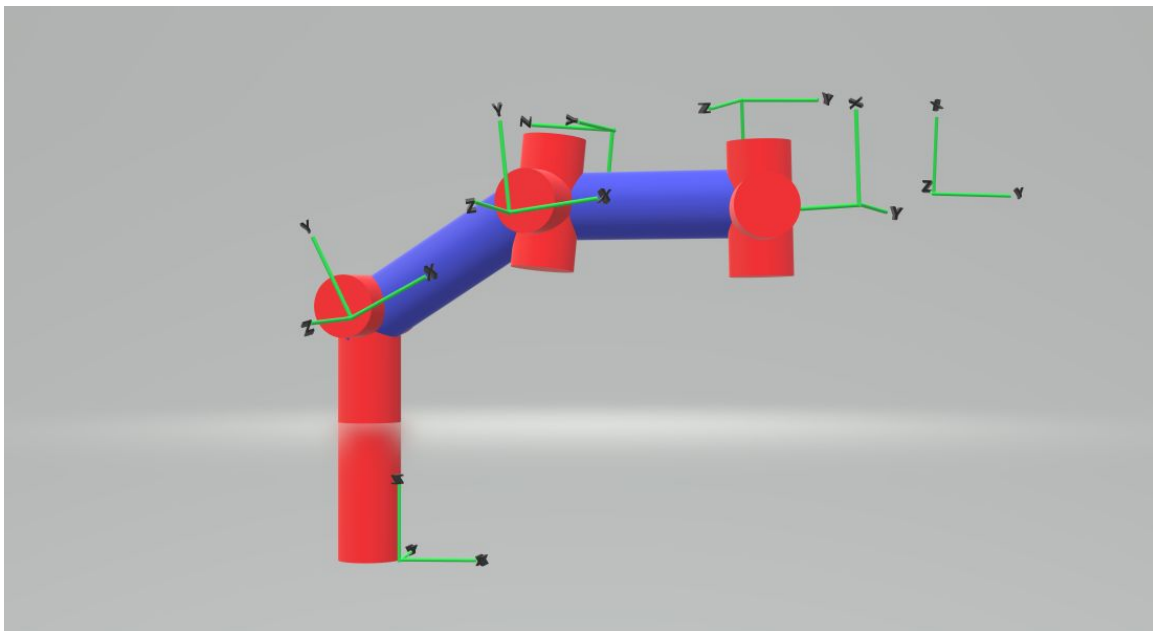
Para este manipulador de propósito general, se emplea seis articulaciones rotacionales. Esto permite controlar todas las piezas directamente mediante la rotación de motores eléctricos, evitando mecanismos para transformar dicho movimiento a uno lineal. Las primeras tres articulaciones permiten posicionar el extremo del robot en cualquier punto del espacio de trabajo, y están separadas por la longitud de las piezas  $L_1$ ,  $L_2$  y  $L_3$ . En el extremo se implementan tres articulaciones más, que se asumen concéntricas (aproximación válida en la

práctica), las cuales implementan los ángulos de Euler ZXZ con el fin de garantizar cualquier orientación posible de la herramienta. A continuación se presenta un diagrama general de la estructura del robot que permite visualizar las piezas, las articulaciones y sus dimensiones.



### **Cinemática directa**

El objetivo de este apartado es encontrar una relación funcional entre las variables articulares  $q_1, q_2, q_3, q_4, q_5$  y  $q_6$  y la posición y orientación del extremo del manipulador. Para ello se definen los sistemas de coordenadas  $\{S_i\}$  (con  $i=0,1,\dots,6$ ) bajo la convención de Denavit-Hartenberg. A continuación se presenta la estructura del robot junto con dichos sistemas.



A partir del diagrama anterior es posible obtener los parámetros de Denavit-Hartenberg para cada articulación, los cuales se presentan en la siguiente tabla.

i	Matriz	$\theta$	d	a	$\alpha$
1	$A_0^1$	$q_1$	$L_1$	0	$\pi/2$
2	$A_1^2$	$q_2$	0	$L_2$	0
3	$A_2^3$	$q_3$	0	0	$\pi/2$
4	$A_3^4$	$q_4$	$-L_3$	0	$-\pi/2$
5	$A_4^5$	$q_5$	0	0	$\pi/2$
6	$A_5^6$	$q_6$	0	0	$-\pi/2$

**Tabla (1)**

Para obtener la expresión deseada, es necesario hallar las matrices de transformación  $A_{i-1}^i$  así como la matriz de transformación final dada por  $T=A_0^1 A_1^2 A_2^3 A_3^4 A_4^5 A_5^6$ . Dichas expresiones se presentan a continuación:

$$A_0^1 = \begin{bmatrix} C_{q1} & 0 & S_{q1} & 0 \\ S_{q1} & 0 & -C_{q1} & 0 \\ 0 & 1 & 0 & L_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_1^2 = \begin{bmatrix} C_{q2} & -S_{q2} & 0 & L_2 C_{q2} \\ S_{q2} & C_{q2} & 0 & L_2 S_{q2} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_2^3 = \begin{bmatrix} C_{q3} & 0 & S_{q3} & 0 \\ S_{q3} & 0 & -C_{q3} & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_3^4 = \begin{bmatrix} C_{q4} & 0 & -S_{q4} & 0 \\ S_{q4} & 0 & C_{q4} & 0 \\ 0 & -1 & 0 & -L_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_4^5 = \begin{bmatrix} C_{q5} & 0 & S_{q5} & 0 \\ S_{q5} & 0 & -C_{q5} & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_5^6 = \begin{bmatrix} C_{q6} & 0 & -S_{q6} & 0 \\ S_{q6} & 0 & C_{q6} & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

El producto de las matrices anteriores permite obtener una matriz de transformación de la forma:

$$\mathbf{T} = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Para el caso de la *posición del extremo del manipulador*, como es de esperarse por la estructura física del robot, está determinada por las variables articulares  $q_1, q_2$  y  $q_3$  de la siguiente manera:[20 15 30 15 12 41]

$$p_x = -L_3 C_{q1} C_{q2} S_{q3} - L_3 C_{q1} S_{q2} C_{q3} + L_2 C_{q1} C_{q2} \quad (1)$$

$$p_y = -L_3 S_{q1} C_{q2} S_{q3} - L_3 S_{q1} S_{q2} C_{q3} + L_2 S_{q1} C_{q2} \quad (2)$$

$$p_z = -L_3 S_{q2} S_{q3} + L_3 C_{q2} C_{q3} + L_2 S_{q2} + L_1 \quad (3)$$

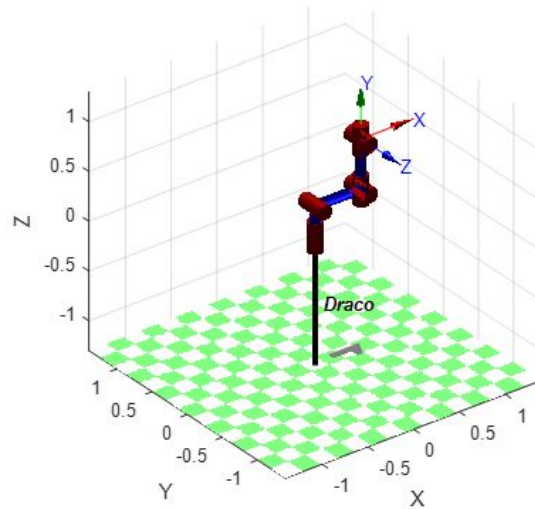
A partir de las ecuaciones anteriores se puede constatar que efectivamente el manipulador posee seis grados de libertad. Mediante las primeras tres variables articulares se puede colocar el extremo del robot en cualquier posición según el sistema no lineal de tres variables e incógnitas definido por las ecuaciones (1), (2) y (3). Por otro lado, una vez definidas las variables anteriores, las ecuaciones de orientación  $\mathbf{n}$ ,  $\mathbf{o}$  y  $\mathbf{a}$  (muy extensas, no presentadas) definen un sistema no lineal de nueve variables y tres incógnitas. Sin embargo, debido a que la matriz de orientación debe cumplir con ciertas restricciones como ortonormalidad de filas y columnas, es posible hallar soluciones definidas para cualquier orientación deseada.

El manejo de dichas ecuaciones puede resultar engorroso, por ello se emplean sólo como referencia para evaluar las capacidades cinemáticas del robot. En la práctica, la cinemática del manipulador se puede definir mediante un software específico para tal propósito, permitiendo evaluar dichas funciones mucho más fácilmente. En particular, la API del Robotics Toolbox de Peter Corke permite definir la cinemática con la información de la tabla (1), automatizando el cálculo de la matriz  $\mathbf{T}$  para variables articulares definidas numéricamente. A continuación se presentan ejemplos particulares para diferentes posiciones de las articulaciones.

Sea  $\mathbf{q} = [q_1 \ q_2 \ q_3 \ q_4 \ q_5 \ q_6]$ , y tomando las dimensiones del manipulador como  $L_1=0.3 \text{ m}$ ,  $L_2=0.5 \text{ m}$  y  $L_3=0.5 \text{ m}$ ; se tienen las siguientes configuraciones:

- Para  $\mathbf{q} = [0 \ 0 \ 0 \ 0 \ 0 \ 0]$  (posición base)

Manipulador orientado con las variables articulares  $\mathbf{q}=[0 \ 0 \ 0 \ 0 \ 0 \ 0]$  (grados)

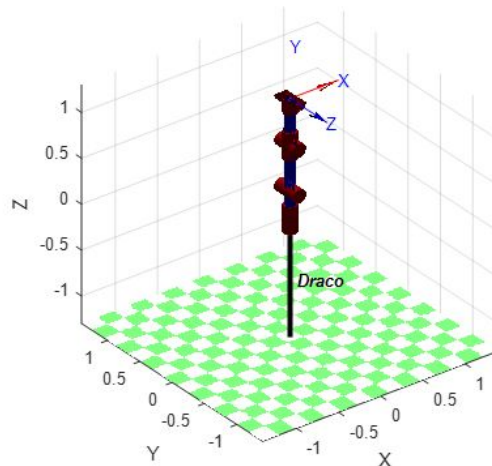


$T=$

1	0	0	0.5
0	0	-1	0
0	1	0	0.8
0	0	0	1

- Para  $\mathbf{q} = [0 \ 90 \ -90 \ 0 \ 0 \ 0]$  (extendido verticalmente)

Manipulador orientado con las variables articulares  $\mathbf{q}=[0 \ 90 \ -90 \ 0 \ 0 \ 0]$  (grados)



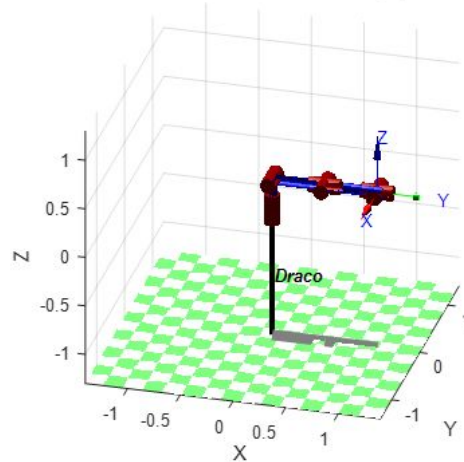


T=

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 1.3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- Para  $\mathbf{q} = [0 \ 0 \ -90 \ 0 \ 0 \ 90]$  (extendido horizontalmente y herramienta rotada perpendicularmente)

Manipulador orientado con las variables articulares  $\mathbf{q}=[0 \ 0 \ -90 \ 0 \ 0 \ 90]$  (grados)



T=

$$\begin{bmatrix} 0 & 1 & 0 & 1 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0.3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

### Cinemática inversa

El objetivo de la cinemática inversa será encontrar expresiones con las cuales, dada una determinada posición del extremo del manipulador, sea posible encontrar el valor de las variables articulares que permitan llegar a esta posición.

A diferencia del caso de la cinemática directa donde había una solución única, en este caso es posible encontrar diversas soluciones.

Se parte de las ecuaciones de la *posición del extremo del manipulador*:

$$p_x = -L_3 C_{q1} C_{q2} S_{q3} - L_3 C_{q1} S_{q2} C_{q3} + L_2 C_{q1} C_{q2} \quad (1)$$

$$p_y = -L_3 S_{q1} C_{q2} S_{q3} - L_3 S_{q1} S_{q2} C_{q3} + L_2 S_{q1} C_{q2} \quad (2)$$

$$p_z = -L_3 S_{q2} S_{q3} + L_3 C_{q2} C_{q3} + L_2 S_{q2} + L_1 \quad (3)$$

Dividiendo (1) entre  $C_{q1}$  y (2) entre  $S_{q1}$  obtenemos:

$$\frac{p_x}{C_{q1}} = -L_3 * C_{q2} * S_{q3} - L_3 * S_{q2} * C_{q3} + L_2 * C_{q2} \quad (4)$$

$$\frac{p_y}{S_{q1}} = -L_3 * C_{q2} * S_{q3} - L_3 * S_{q2} * C_{q3} + L_2 * C_{q2} \quad (5)$$

Luego, tomando (4) – (5) y despejando:

$$\frac{p_x}{C_{q1}} = \frac{p_y}{S_{q1}} \quad (6)$$

$$\tan q_1 = \frac{p_y}{p_x}$$

De donde:

$$q_1 = \tan^{-1} \frac{p_y}{p_x}$$

Para despejar  $q_2$  se parte nuevamente de (4) y (5):

$$\frac{p_x}{C_{q1}} - L_2 * C_{q2} = \frac{p_y}{S_{q1}} - L_2 * C_{q2}$$

Elevando al cuadrado la expresión anterior:

$$\left( \frac{p_x}{C_{q1}} - L_2 * C_{q2} \right)^2 = \left( \frac{p_y}{S_{q1}} - L_2 * C_{q2} \right)^2$$

$$\left( \frac{p_x}{C_{q1}} \right)^2 - 2 * \frac{p_x}{C_{q1}} * L_2 * C_{q2} + (L_2 * C_{q2})^2 = \left( \frac{p_y}{S_{q1}} \right)^2 - 2 * \frac{p_y}{S_{q1}} * L_2 * C_{q2} + (L_2 * C_{q2})^2$$

$$\begin{aligned}\left(\frac{p_x}{C_{q1}}\right)^2 - 2 * \frac{p_x}{C_{q1}} * L_2 * C_{q2} &= \left(\frac{p_y}{S_{q1}}\right)^2 - 2 * \frac{p_y}{S_{q1}} * L_2 * C_{q2} \\ \left(\frac{p_x}{C_{q1}}\right)^2 - \left(\frac{p_y}{S_{q1}}\right)^2 &= 2 * L_2 * C_{q2} * \left(\frac{p_x}{C_{q1}} - \frac{p_y}{S_{q1}}\right) \\ \left(\frac{p_x}{C_{q1}} + \frac{p_y}{S_{q1}}\right) &= 2 * L_2 * C_{q2}\end{aligned}$$

De donde, usando (6):

$$\begin{aligned}\frac{p_x}{C_{q1}} &= L_2 * C_{q2} \\ q_2 &= \cos^{-1}\left(\frac{p_x}{C_{q1} * L_2}\right)\end{aligned}$$

Esta expresión es válida para el caso en que:

$$C_{q2} * S_{q3} = S_{q2} * C_{q3}$$

No fue posible despejar manualmente una expresión de  $q_2$  válida para todos los casos.

Finalmente, para obtener  $q_3$  se despeja (3):

$$\begin{aligned}\frac{p_z - L_1 - L_2 * S_{q2}}{L_3} &= -S_{q2} * S_{q3} + C_{q2} * C_{q3} \\ \frac{p_z - L_1 - L_2 * S_{q2}}{L_3} &= \cos(q_2 + q_3)\end{aligned}$$

De donde:

$$q_3 = \cos^{-1}\left(\frac{p_z - L_1 - L_2 * S_{q2}}{L_3}\right) - q_2$$

Este proceso es engorroso e involucra muchos cálculos. Usando el Robotics Toolbox es posible realizar la cinemática inversa de una forma muy sencilla. Se comienza creando una matriz de transformación Tf usando la expresión:

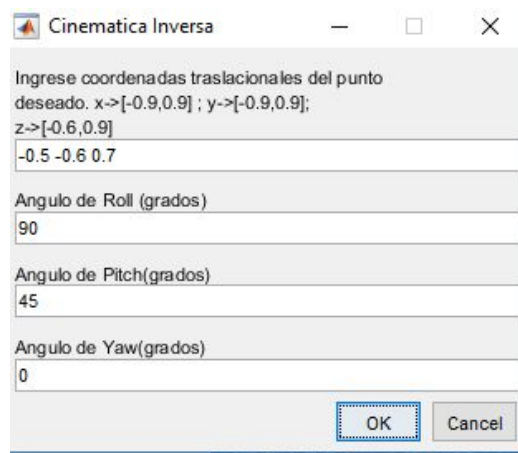
$$Tf = SE3(tf) * SE3.Rz(qr) * SE3.Ry(qp) * SE3.Rx(qy);$$

En donde “ $t_f$ ” son las coordenadas del extremo del manipulador y “ $q_r$ ,  $q_p$ ,  $q_y$ ” son los ángulos de “roll, pitch, yaw” respectivamente, en los cuales se desea ubicar el manipulador.

Luego se usa la función “ $\text{ikine}(T_f)$ ” la cual efectúa de forma numérica el cálculo de la cinemática inversa a partir de una matriz de transformación y arroja como resultado un vector “ $q_f$ ” que contiene los valores de las variables articulares:

```
qf = draco.ikine(Tf)
```

A continuación, se muestra un ejemplo del funcionamiento de este módulo del programa. Se introducen los valores de posición y orientación deseados:



Cinemática Inversa

Ingrese coordenadas traslacionales del punto deseado. x->[-0.9,0.9] ; y->[-0.9,0.9]; z->[-0.6,0.9]

-0.5 -0.6 0.7

Angulo de Roll (grados)

90

Angulo de Pitch(grados)

45

Angulo de Yaw(grados)

0

OK Cancel

Y el programa ubica al manipulador y muestra sus respectivas coordenadas

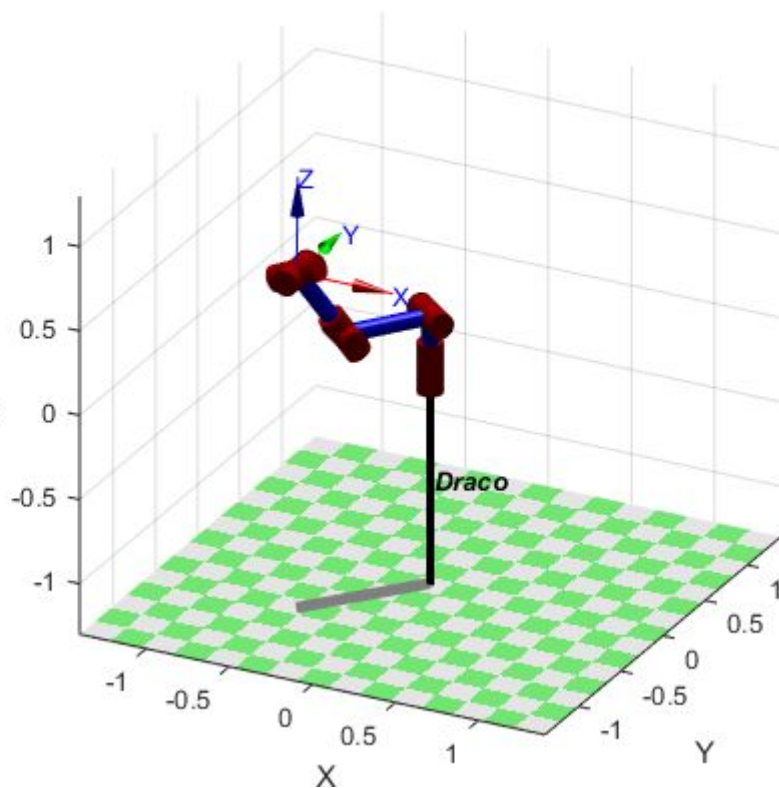
Teach

X: -0.500  
Y: -0.600  
Z: 0.700

R: 1.571  
P: 0.785  
Y: -0.022

q1			-130
q2			-1.54
q3			-32.7
q4			133
q5			-115
q6			65

Manipulador en la posición deseada.



En otra posición:

Cinematica Inversa

Ingrese coordenadas traslacionales del punto deseado. x->[-0.9,0.9] ; y->[-0.9,0.9]; z->[-0.6,0.9]

0.3 0.4 -0.5

Angulo de Roll (grados)

45

Angulo de Pitch(grados)

0

Angulo de Yaw(grados)

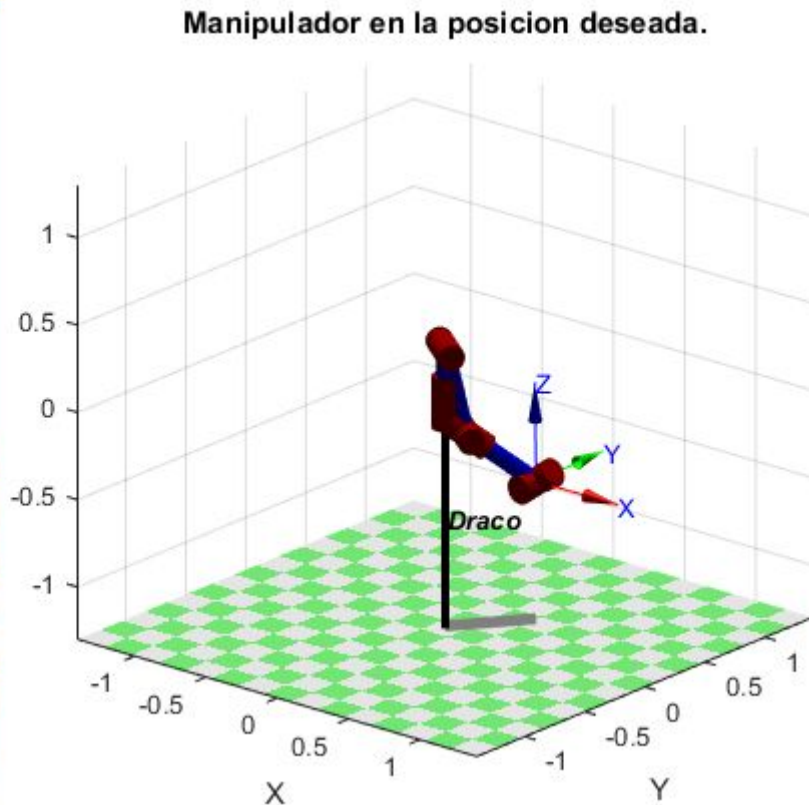
60

Teach

X: 0.300  
Y: 0.400  
Z: -0.500

R: 0.785  
P: 0.014  
Y: 1.047

q1	◀		▶	53.1
q2	◀		▶	-77.4
q3	◀		▶	-51.3
q4	◀		▶	50.2
q5	◀		▶	52.8
q6	◀		▶	53.1



### Generación de Trayectorias

Para definir una trayectoria se empleará el método de interpolación polinómica, que en el Toolbox viene dada por la función `mtraj`:

```
[q qd qdd] = mtraj(@tpoly,qi, qf, t);
```

En donde sus argumentos son:

`qi` , `qf`: posiciones articulares inicial y final respectivamente.

`t`: vector de tiempo

`Y` genera:

`q`: vector con la interpolación de posiciones articulares.

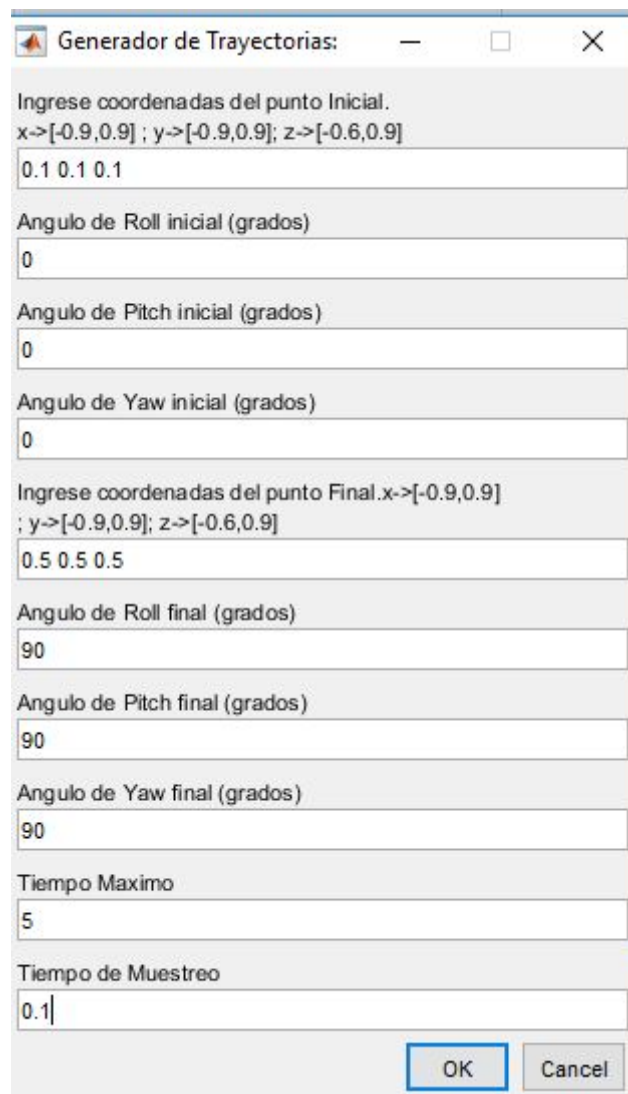
`qd`: vector de velocidad

`qdd`: vector de aceleración

El comando @tpoly le indica a la función que ejecute la interpolación de forma polinómica. Cabe mencionar que está disponible un comando @lspb para generar trayectorias LSPB sin embargo, se probó su funcionamiento y se obtuvieron resultados idénticos al caso polinómico.

A continuación, se muestra un ejemplo del funcionamiento de este módulo del programa.

Se solicitan los siguientes parámetros de entrada:



Generador de Trayectorias: — □ ×

Ingrese coordenadas del punto Inicial.  
x->[-0.9,0.9] ; y->[-0.9,0.9]; z->[-0.6,0.9]

0.1 0.1 0.1

Angulo de Roll inicial (grados)

0

Angulo de Pitch inicial (grados)

0

Angulo de Yaw inicial (grados)

0

Ingrese coordenadas del punto Final.x->[-0.9,0.9]  
; y->[-0.9,0.9]; z->[-0.6,0.9]

0.5 0.5 0.5

Angulo de Roll final (grados)

90

Angulo de Pitch final (grados)

90

Angulo de Yaw final (grados)

90

Tiempo Maximo

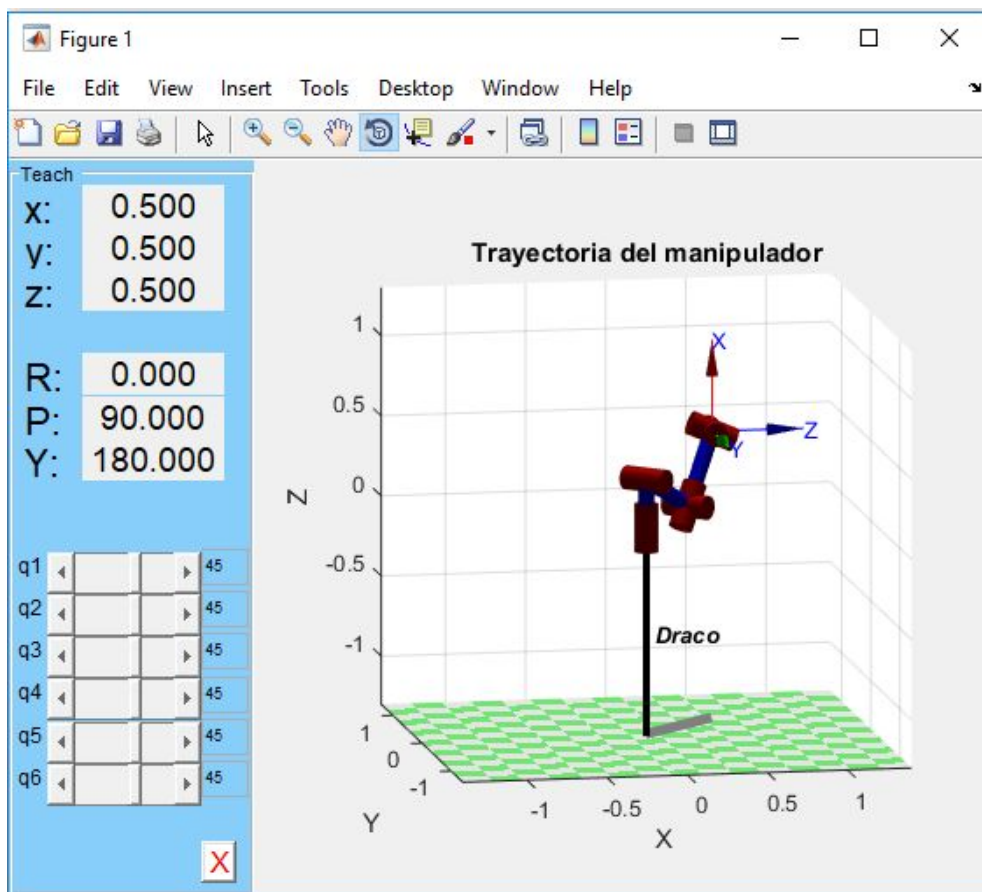
5

Tiempo de Muestreo

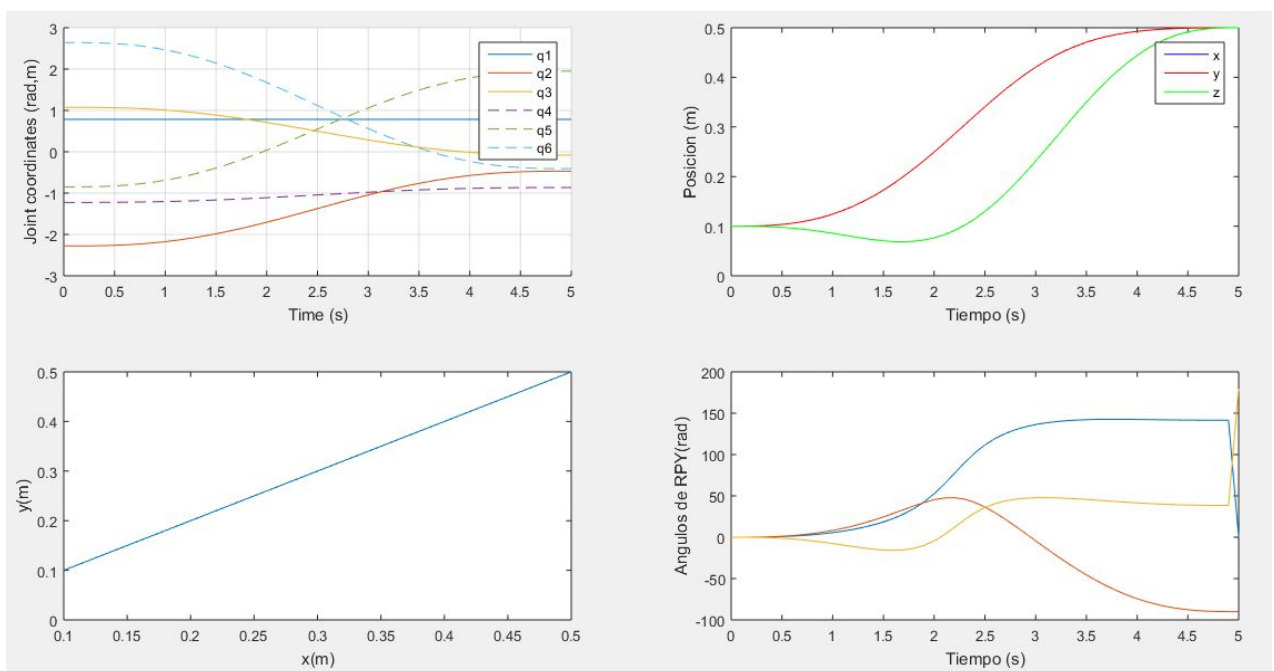
0.1

OK Cancel

Se muestra una animación con la trayectoria definida:



Y además se generan las siguientes gráficas:





## Velocidad cinemática

En esta sección se estudia la relación funcional entre la velocidad del extremo del manipulador y sus variables articulares. Así mismo, se estudia sus posibles configuraciones singulares. Para ello, es necesario encontrar la matriz jacobiana del robot, la cual se obtiene a partir de las funciones halladas en la cinemática directa. Debido a la alta complejidad de las funciones que describen la orientación del manipulador, solo se analizará teóricamente su posición. A continuación se halla el jacobiano asociado a la posición.

Partiendo de las ecuaciones halladas en la sección anterior:

- Gradiente de la coordenada cartesiana x:

$$p'_x(q_1) = L_3 S_{q1} C_{q2} S_{q3} + L_3 S_{q1} S_{q2} C_{q3} - L_2 S_{q1} C_{q2}$$

$$p'_x(q_2) = L_3 C_{q1} S_{q2} S_{q3} - L_3 C_{q1} C_{q2} C_{q3} - L_2 C_{q1} S_{q2}$$

$$p'_x(q_3) = -L_3 C_{q1} C_{q2} C_{q3} + L_3 C_{q1} S_{q2} S_{q3}$$

- Gradiente de la coordenada cartesiana y:

$$p'_y(q_1) = -L_3 C_{q1} C_{q2} S_{q3} - L_3 C_{q1} S_{q2} C_{q3} + L_2 C_{q1} C_{q2}$$

$$p'_y(q_2) = L_3 S_{q1} S_{q2} S_{q3} - L_3 S_{q1} C_{q2} C_{q3} - L_2 S_{q1} S_{q2}$$

$$p'_y(q_3) = -L_3 S_{q1} C_{q2} C_{q3} + L_3 S_{q1} S_{q2} S_{q3}$$

- Gradiente de la coordenada cartesiana z:

$$p'_z(q_1) = 0$$

$$p'_z(q_2) = -L_3 C_{q2} S_{q3} - L_3 S_{q2} C_{q3} + L_2 C_{q2}$$

$$p'_z(q_3) = -L_3 S_{q2} C_{q3} - L_3 C_{q2} S_{q3}$$

Finalmente se obtiene el jacobiano de posición como:

$$J_p = \begin{bmatrix} L_3 S_{q1} C_{q2} S_{q3} + L_3 S_{q1} S_{q2} C_{q3} - L_2 S_{q1} C_{q2} & L_3 C_{q1} S_{q2} S_{q3} - L_3 C_{q1} C_{q2} C_{q3} - L_2 C_{q1} S_{q2} & -L_3 C_{q1} C_{q2} C_{q3} + L_3 C_{q1} S_{q2} S_{q3} \\ -L_3 C_{q1} S_{q2} S_{q3} - L_3 C_{q1} S_{q2} C_{q3} + L_2 C_{q1} C_{q2} & -L_3 S_{q1} C_{q2} S_{q3} - L_3 S_{q1} C_{q2} C_{q3} - L_2 S_{q1} S_{q2} & -L_3 S_{q1} S_{q2} C_{q3} + L_3 S_{q1} S_{q2} S_{q3} \\ 0 & -L_3 C_{q2} S_{q3} - L_3 S_{q2} C_{q3} + L_2 C_{q2} & -L_3 S_{q2} C_{q3} - L_3 C_{q2} S_{q3} \end{bmatrix}$$

Para estudiar a mayor profundidad la movilidad del robot, se realiza dos tipos de análisis numéricos que permiten determinar la manipulabilidad del mismo en una configuración particular o una trayectoria específica. Si el análisis de una posición en particular revela lo siguiente:

- **Matriz jacobiana:** la relación funcional entre  $\mathbf{V}'$  y  $\mathbf{q}'$  (velocidades del extremo del manipulador y de las articulaciones respectivamente).
- **Manipulabilidad:** valor real asociado a la capacidad de movilidad del robot para una configuración dada. Su valor es cero si el punto es singular. Se calcula como:

$$m = \sqrt{\det(JJ^T)}$$

- **Gráficas de las posibles ternas de velocidades traslacionales y rotacionales:** son todas las posibles combinaciones de velocidades posibles para el robot en una configuración dada. Para cada caso (traslacional y rotacional), recrea una elipsoide. La movilidad aumenta a medida que dicha elipsoide se parece más a una esfera.

A continuación se analizan varias posiciones notables mediante el software desarrollado.

- Manipulador en  $\mathbf{q}=[0 \ 0 \ 0 \ 0 \ 0 \ 0]$ :

Esta configuración ya fue graficada previamente. Analizando la movilidad para este caso, se obtiene lo siguiente.

*Matriz jacobiana:*

$$J = \begin{bmatrix} -0.0000 & -0.5000 & -0.5000 & 0 & 0 & 0 \\ 0.5000 & 0.0000 & 0 & 0 & 0 & 0 \\ 0 & 0.5000 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1.0000 & -1.0000 & -0.0000 & -1.0000 & -0.0000 \\ 1.0000 & 0.0000 & 0.0000 & -1.0000 & 0.0000 & -1.0000 \end{bmatrix}$$

Es fácil notar que la determinante es cero ya que una fila es nula. Es decir, esta posición es singular.

*Manipulabilidad:*

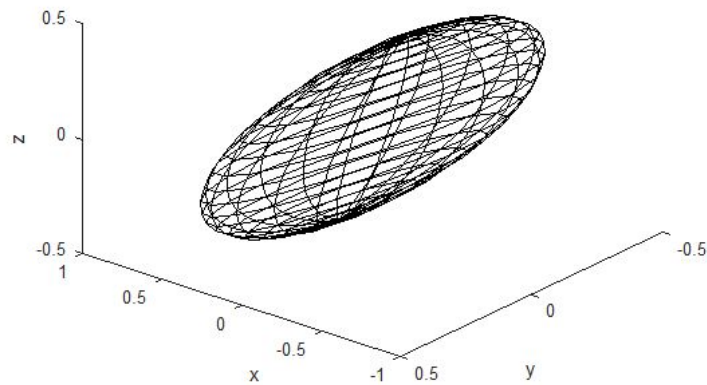
Traslacional: 0.125

Rotacional: 0

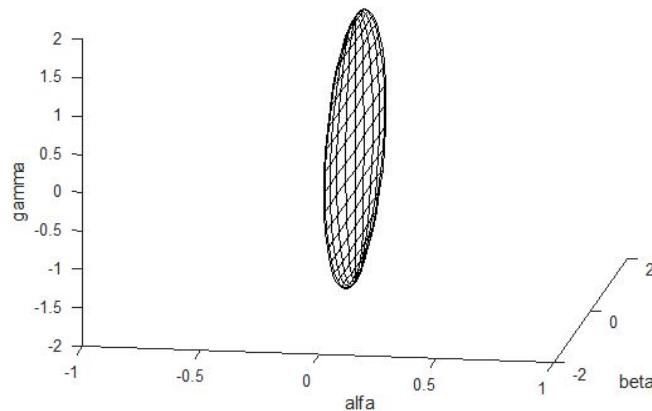
Esto implica que la posición es singular para la orientación de la muñeca, mas no para la posición del extremo del manipulador.

### *Posibles posiciones*

**Posibles velocidades traslacionales**



**Posibles velocidades rotacionales**



Es interesante observar que la velocidad angular en alfa es siempre cero, sin importar el valor de las variables articulares. Esto es consecuencia de la posición singular: es necesaria una variación infinita en  $q_i$  para lograr tener una velocidad angular en alfa distinta de cero (imposible). Esto último no ocurre para la traslación, en el cual se observa un elipsoide.

- Manipulador en  $q=[0 \ 45 \ 180 \ 0 \ 45 \ 0]$ :

## Jacobiano

$J =$

-0.0000	0.0000	0.3536	0	0	0
0.7071	0.0000	0.0000	0	0	0
0.0000	0.7071	0.3536	0	0	0
-0.0000	-0.0000	-0.0000	-0.7071	-0.0000	-1.0000
-0.0000	-1.0000	-1.0000	-0.0000	-1.0000	-0.0000
1.0000	0.0000	0.0000	0.7071	0.0000	0.0000

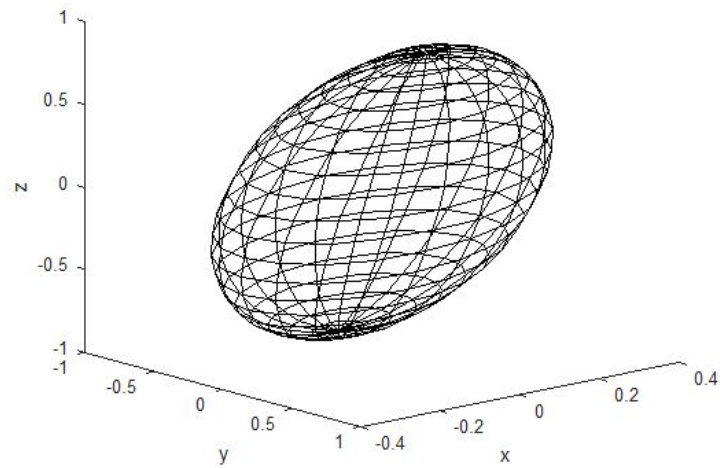
## Manipulabilidad

Traslacional: 0.176777

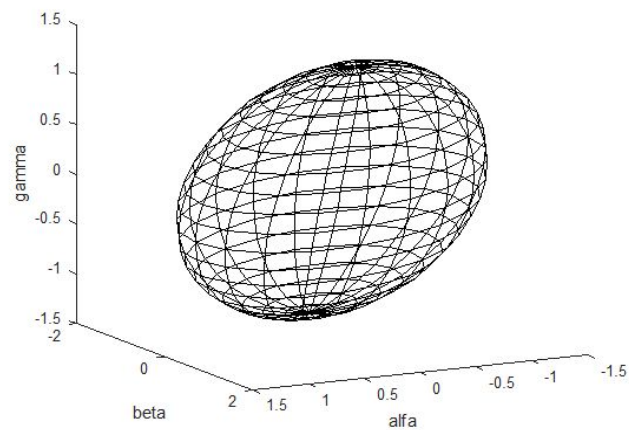
Rotacional: 2.44949

## Posibles posiciones

Posibles velocidades translacionales

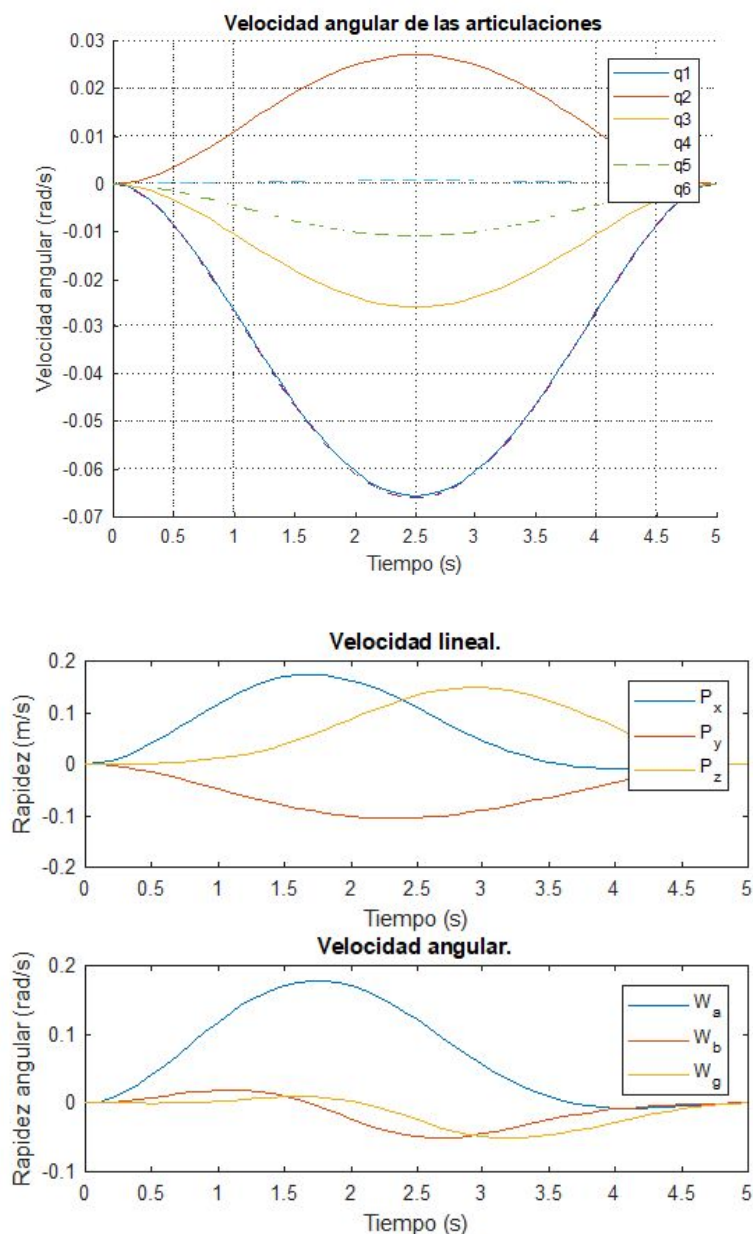


Posibles velocidades rotacionales



Por otro lado, si se emplea la segunda opción planteada en el programa, es posible analizar una trayectoria previamente construida. En la sección de planificador de trayectorias se detalla cómo generar un camino en particular. A continuación se presenta un ejemplo del análisis de movilidad de una trayectoria.

- Velocidades para una trayectoria generada desde la posición de la herramienta  $[0.1 \ 0.2 \ 0.3 \ 0 \ 0 \ 0]$  hasta  $[0.4 \ -0.3 \ 0.5 \ \pi/2 \ \pi/2 \ \pi/2]$  en un tiempo de 5 segundos.



## Dinámica del manipulador

El objetivo de esta sección es hallar una relación funcional entre las posiciones, velocidades y aceleraciones articulares y los torques necesarios para lograrlos. Esto se conoce como el modelo dinámico del robot. El modelo directo halla la configuración cinemática para un vector de torques dado, y el modelo inverso lo contrario. Uno de los métodos más usados para modelar la dinámica es el lagrangiano, el cual emplea las energías potencial y cinética para estructurar las ecuaciones el movimiento. Sin embargo, la complejidad de estos modelos aumenta considerablemente conforme a la cantidad de grados de libertad del manipulador, de modo que para seis grados se tiene relaciones matemática altamente extensas. Para nuestro caso es mucho más conveniente y eficiente emplear métodos numéricos basados en el algoritmo de Newton-Euler, el cual empieza desde la base y añade la velocidad y aceleración de cada articulación para determinar las velocidades y aceleraciones de cada eslabón.

El modelo dinámico del manipulador puede ser descrito por la siguiente ecuación matricial diferencial:

$$\mathbf{Q} = \mathbf{M}(\mathbf{q})\mathbf{q}'' + \mathbf{C}(\mathbf{q}, \mathbf{q}')\mathbf{q}' + \mathbf{F}(\mathbf{q}') + \mathbf{G}(\mathbf{q}) + \mathbf{J}(\mathbf{q})^T \mathbf{W}$$

Donde  $\mathbf{M}(\mathbf{q})$  es la matriz de momento de inercial de las articulaciones,  $\mathbf{C}(\mathbf{q}, \mathbf{q}')$  es la matriz de coriolis y centrípeta,  $\mathbf{F}(\mathbf{q}')$  la matriz de fricción,  $\mathbf{G}(\mathbf{q})$  la carga gravitatoria,  $\mathbf{J}(\mathbf{q})$  el jacobiano,  $\mathbf{W}$  el vector de carga en el extremo del manipulador y  $\mathbf{Q}$  el vector de torques de las articulaciones. Todas estas matrices pueden ser evaluadas numéricamente mediante el algoritmo de Newton-Euler para una configuración de  $\mathbf{q}$ ,  $\mathbf{q}'$  y  $\mathbf{q}''$  dada.

Previamente a evaluar la ecuación anterior, es necesario definir los parámetros dinámicos del manipulador. Estos se definen a continuación por eslabón. Recordemos que  $L_1=0.3$  m,  $L_2=0.5$  m y  $L_3=0.5$  m. Se define el ancho y grueso de todos los eslabones como  $w=0.04$  y  $t=0.02$ . Se consideran todas las piezas como paralelepípedos.

Los siguientes valores fueron calculados a partir de las dimensiones dadas, una masa máxima supuesta por pieza, y referente al sistema de coordenadas j fijo para cada eslabón (todo en unidades MSK).

- Eslabón 1:  
 $m=3.5$  kg (masa del eslabón)  
 $I=[0.0264 \ 0.0006 \ 0.0267 \ 0 \ 0 \ 0]$  (tensor del centro de masas)  
 $r=[0 \ -0.15 \ 0]$  (posición del centro de masas respecto al sc j)
- Eslabón 2:  
 $m=5.5$  kg  
 $I=[0.0009 \ 0.1148 \ 0.1153 \ 0 \ 0 \ 0]$   
 $r=[-0.25 \ 0 \ 0]$
- Eslabón 3:  
 $m=4$  kg  
 $I=[0.0835 \ 0.0839 \ 0.0007 \ 0 \ 0 \ 0]$   
 $r=[0 \ 0 \ -0.25]$
- Eslabón 4,5,6: son los de orientación, se desprecia su masa e inercia.  
 $m=0.25$  kg  
 $I=[0.3e-3 \ 0.4e-3 \ 0.3e-3 \ 0 \ 0 \ 0]$   
 $r=[0 \ 0 \ 0]$

Para todos los motores (de las articulaciones 1,2 y 3), se consideran los siguientes parámetros:

$J_m=250 \cdot 10^{-6}$  (inercial del actuador)

$B=1.2 \cdot 10^{-3}$  (coeficiente de fricción viscosa)

$T_{c_i}=[0.218 \ -0.200]$  (coeficiente de fricción de coulomb por dirección)

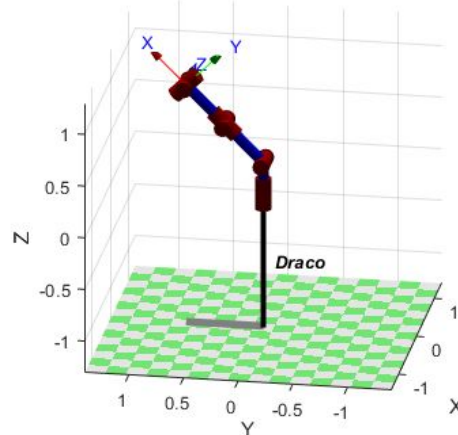
$G=85$  (relación de los engranajes).

El software desarrollado permite dos opciones de análisis similares al caso de la movilidad. La primera permite analizar una configuración en particular, es decir para una posición, velocidad y aceleración definida para todas las articulaciones. Esta calcula: el vector de torques necesarios en cada articulación para alcanzar dicha configuración, la matriz de inercia total, la matriz de coriolis total, el vector de torques generados por la gravedad y el vector de fricción. Esta opción es útil para

estudiar las capacidades dinámicas del robot en casos particulares. A continuación se presenta un ejemplo.

- Dinámica del manipulador para:  $q=[90 \ 45 \ -90 \ 0 \ 90 \ 0]$ ,  $q'=[45 \ 10 \ 30 \ 45 \ 0 \ 45]$ ,  $q''=[0 \ 15 \ 10 \ -5 \ 0 \ -5]$  y sosteniendo una masa de 1 kg.

Manipulador orientado con las variables articulares  $q=[90 \ 45 \ -90 \ 0 \ 90 \ 0]$  (grados)



Al emplear el programa en la opción de dinámica se obtiene:

*Matriz de inercia:*

```
M_inertia =
    3.3805    0.0000    0.0000    0.0629    0.0000   -0.0003
    0.0000    5.6098    1.2195    0.0000    0.0107    0.0000
    0.0000    1.2195    2.6508    0.0000    0.0107    0.0000
    0.0629    0.0000    0.0000    0.0110    0.0000    0.0000
    0.0000    0.0107    0.0107    0.0000    0.0107    0.0000
   -0.0003    0.0000    0.0000    0.0000    0.0000    0.0004
```

Se puede apreciar que el elemento mayor es el segundo de la diagonal, el cual corresponde a la inercia de la segunda articulación. Esto es concordante con la posición observada, ya que al tener el brazo extendido es de esperarse tener una mayor inercia.

*Matriz de coriolis*

```
C_coriolis =
   -0.5168   -1.1251   -0.4005   -0.0002   -0.0350    0.0001
    1.1861   -0.3862   -0.5149   -0.0170   -0.0698    0.0000
    0.4338    0.1287    0.0000   -0.0055   -0.0349    0.0000
   -0.0360   -0.0615   -0.0338   -0.0000   -0.0059   -0.0001
    0.0410    0.0436    0.0349    0.0059   -0.0000    0.0000
    0.0001    0.0003    0.0003    0.0001   -0.0000    0.0000
```



*Vector de torques necesarios para lograr la configuración planteada (N\*m):*

**[24.5219    51.0934    19.1813    -0.0577856    -0.626264    0.000339504]**

Como se puede apreciar, el torque necesario en la segunda articulación es significativamente mayor que en el resto, lo cual es de esperarse según lo discutido anteriormente.

*Vector de torques necesarios únicamente para contrarrestar la gravedad (velocidad y aceleración cero) (N\*m):*

**[0    28.7874    -5.02912    -1.02544e-16    -0.693672    -4.24751e-17]**

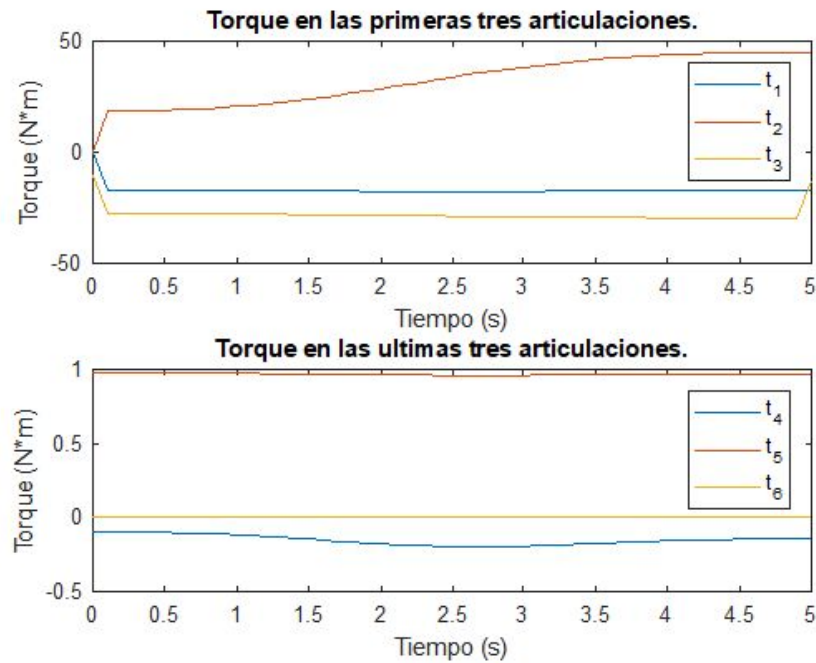
*Vector de torques de fricción de las articulaciones para la velocidad dada (N\*m):*

**[-25.3394    -20.0432    -23.0696    0    0    0]**

Es importante destacar el efecto significativo que tiene la gravedad y la fricción entre las articulaciones, los cuales afectan significativamente el vector de torques necesarios para lograr el movimiento deseado. Al ser estos dos factores críticos en la demanda de potencia a los motores, es necesario afinar el diseño mecánico para reducir al mínimo su repercusión en el torque necesario en las articulaciones.

La segunda opción de análisis permite estudiar una trayectoria previamente generada y verificada no singular mediante un análisis de movilidad (leer sección anterior). A partir de la trayectoria, se obtiene la evolución temporal de los torques de cada articulación, lo cual es especialmente útil para el control del torque de los actuadores. A continuación se muestra un ejemplo de este análisis.

- Torques para una trayectoria generada desde la posición de la herramienta [0.1 0.2 0.3 0 0 0] hasta [0.4 -0.3 0.5  $\pi/2$   $\pi/2$   $\pi/2$ ] en un tiempo de 5 segundos y sosteniendo un objeto de masa 1 kg.



### **Pruebas generales**

Tal como se dijo anteriormente, los manipuladores universales son ampliamente usados en el sector industrial. Como una prueba final de las funcionalidades del programa desarrollado, en la opción 7 del menú se encuentra la tarea “Movimiento de productos”. Allí se genera automáticamente una trayectoria sobre la cual se pueden efectuar todos los análisis disponibles.

Esta trayectoria simula al manipulador moviendo productos de un lado a otro de una línea de producción, tal como podría ser usado un robot con estas características en una industria real.

Para hacer más realista el estudio, en el análisis dinámico (opción 6) es posible ubicar una carga del peso seleccionado en el extremo del manipulador y obtener las gráficas con los torques respectivos efectuados por el robot en una trayectoria.

A continuación, se muestran las gráficas obtenidas:

Teach

X: 0.500

Y: -0.500

Z: -0.200

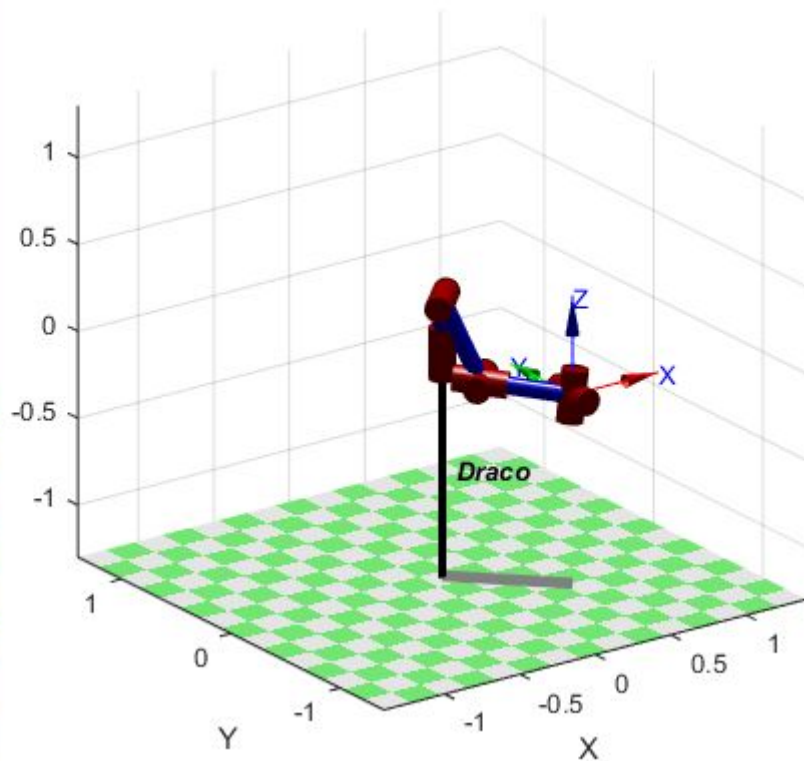
R: -0.000

P: 0.000

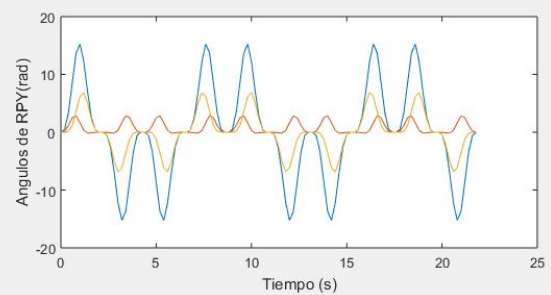
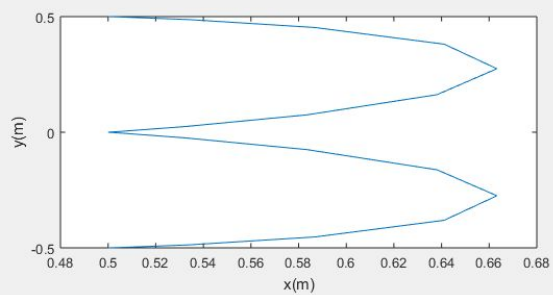
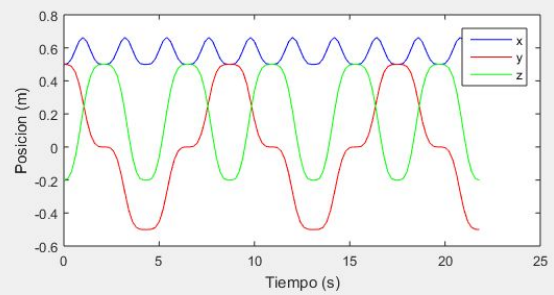
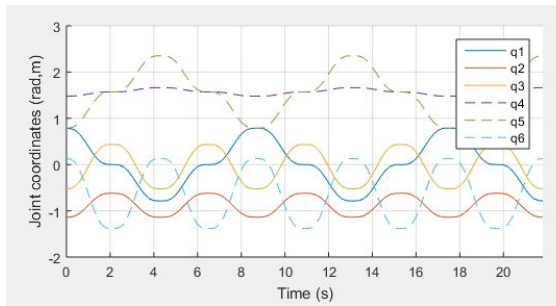
Y: 0.000

q1			45
q2			44.8
q3			42.4
q4			37.7
q5			30.7
q6			22.5

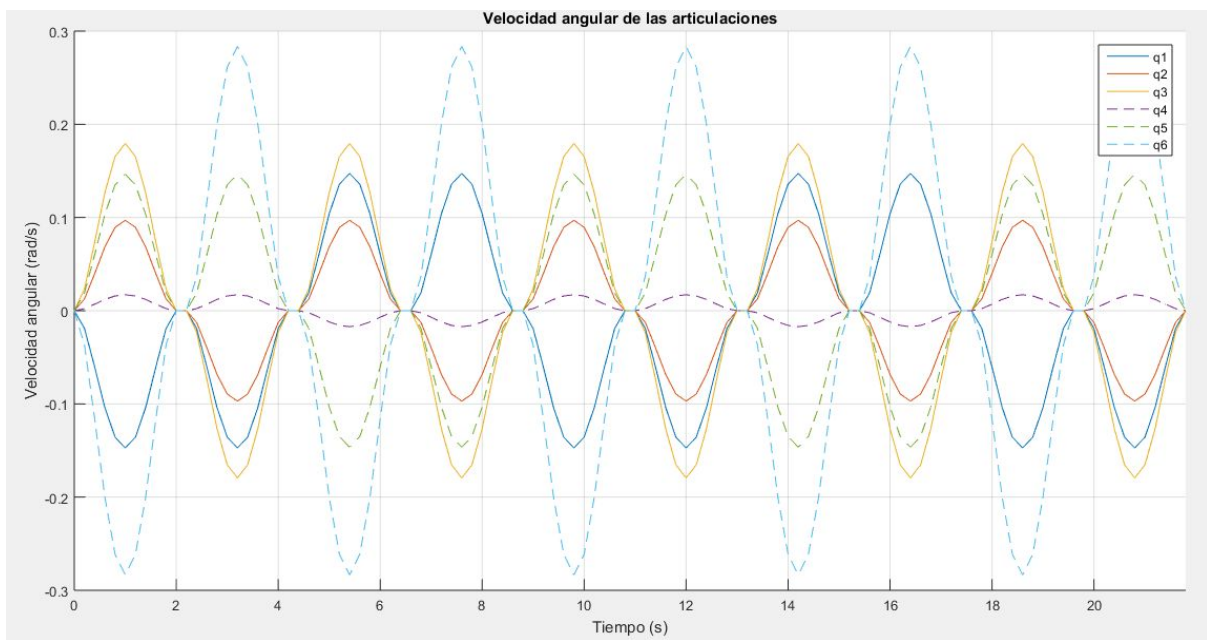
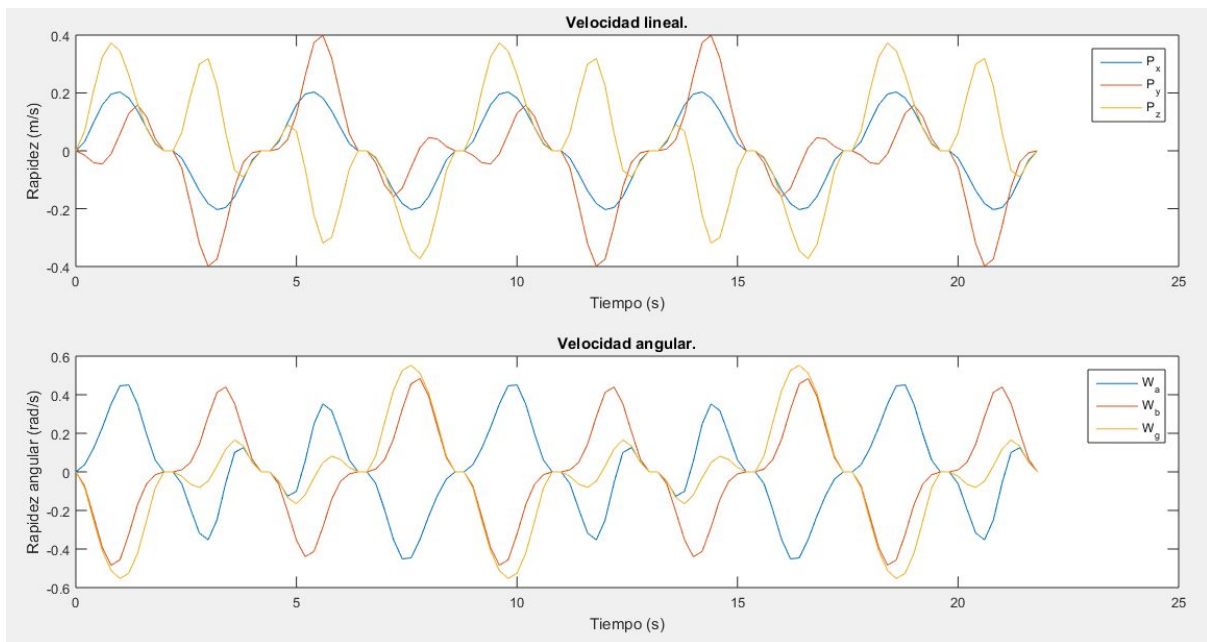
## Trayectoria del manipulador



## Trayectorias:

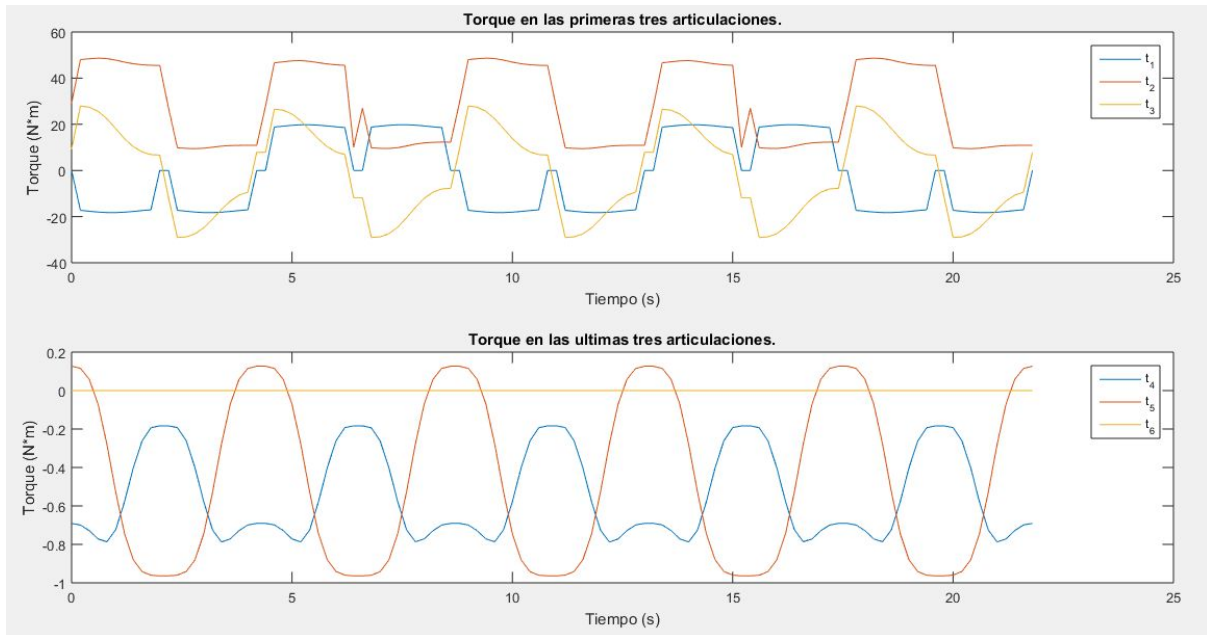


## Velocidad cinemática:



Dinámica:

Sosteniendo un cuerpo de 1Kg en el extremo del manipulador.



## **Conclusiones**

El presente trabajo permitió plantear la estructura de un manipulador universal y estudiar su comportamiento a nivel de cinemática y dinámica. Las dimensiones del robot planteado (largo total de 1.3 m) son similares a las del brazo humano, lo que lo hace ideal para tareas industriales de manufactura en general. Para su estudio se desarrolló un programa en MATLAB capaz de simular trayectorias y realizar un conjunto de análisis sobre la misma. A continuación se repasa brevemente los objetivos logrados.

Primeramente se planteó la estructura física del manipulador, en la cual se definieron los grados de libertad del mismo mediante la convención de Denavit-Hartenberg. Se definió un robot de seis grados de libertad con articulaciones rotacionales. Posteriormente, se realizó el análisis cinemático completo, en la cual se obtuvo las relaciones entre las variables articulares espaciales y la posición del extremo del manipulador.

A partir de lo anterior, se analizó en la medida de lo posible la cinemática inversa. Teóricamente se encontró ecuaciones restringidas, sin embargo el principal análisis completo se realizó numéricamente, de modo que para cualquier punto el programa nos permitió hallar las variables articulares necesarias y graficar tal configuración. Con base a esto, se implementó un planificador de trayectorias polinómicas, el cual parte de la cinemática inversa de un punto final e inicial para encontrar la descripción temporal completa de una trayectoria.

Posteriormente, se crearon módulos para analizar la movilidad del robot y su dinámica. Esto se puede hacer con una configuración particular ( $q, q', q''$  dados), o para una trayectoria generada. De este modo, una vez creada una trayectoria, esta se puede analizar en la parte de movilidad para determinar si es o no singular, y posteriormente se pueden graficar las velocidades y torques en todas las articulaciones y el extremo del robot.

Finalmente, se crea una trayectoria particular que describe una tarea asociada a una línea de producción, en la cual se pueden realizar todos los análisis anteriores. Esto permite observar el desempeño del manipulador directamente en una aplicación real del mismo.

Como se puede notar, se logró plantear la estructura general del manipulador y analizar todos los aspectos fundamentales del mismo, cumpliendo de este modo con los objetivos planteado al inicio. Para etapas posteriores del diseño es necesario indagar a mayor profundidad en la estructura mecánica y la implementación eléctrica de los actuadores y su control; sin embargo para nuestro estudio de alto nivel los resultados obtenidos son satisfactorios y cumplen con el objetivo general propuesto.

### Anexo: ejecución del programa.

Para iniciar el programa ejecute en la terminal de terminal de MATLAB el archivo. “draco\_main.m”. En la consola se desplegará un menú que le permitirá ejecutar los distintos análisis disponibles.

```
Command Window
>> draco_main
Bienvenidos a DracoBot, ¿Qué desea hacer?:
1 = Cinemática Directa
2 = Cinemática Inversa RPY
3 = Generar trayectorias.
4 = Graficar trayectorias.
5 = Velocidad cinemática.
6 = Análisis dinámico
7 = Cargar una tarea.
Otro = Finalizar
fx >
```

Introduzca el número correspondiente a la opción de su preferencia y presione “enter”.

Introduzca los datos solicitados en el formato solicitado, por ejemplo:

```
>> draco_main
Bienvenidos a DracoBot, ¿Qué desea hacer?:
1 = Cinemática Directa
2 = Cinemática Inversa RPY
3 = Generar trayectorias.
4 = Graficar trayectorias.
5 = Velocidad cinemática.
6 = Análisis dinámico
7 = Cargar una tarea.
Otro = Finalizar
>1.
fx Ingrese el vector de variables articulares en grados (Ejm: [0 0 0 0 0 30]):[0 0 0 0 0 30]
```

El programa le indicará los datos que necesita introducir en cada caso, así como también los pasos necesarios para ejecutar cada análisis.



## **Referencias**

Corke, P. (2017) Robotics, Vision and Control FUNDAMENTAL ALGORITHMS IN MATLAB®. Switzerland: Springer