Universidad de Salamanca | Grado en Ingeniería Informática

# Manipuladores

## Parámetros y Transformaciones de Denavit-Hartenberg

López Sánchez, Javier
Mateos Pedraza, Alejandro

**Abstract:** This document is intended to provide a brief demonstration of how the Denavit-Hartenberg parameters and transformations work when calculating the end position of a handling robot.

**Keywords:** Handling Robot, Robotics, Denavit-Hartenberg, Matlab, CoppeliaSim

## 1. Construcción del Robot: CoppeliaSim

Haciendo uso del simulador CoppeliaSim se ha confeccionado un robot con seis grados de libertad (6-DOF), excluyendo la articulación empleada para ubicar la punta, que necesitaremos para realizar mediciones de la posición final en función de los ángulos de rotación de cada articulación. Es preciso hacer una recopilación de las diferentes piezas y articulaciones de que se compone el manipulador, a fin de obtener un acercamiento más preciso a sus componentes y posibilidades de movimiento:

**Relación de piezas:**

| Nombre | Tipo de pieza | Dimensiones |
|---|---|---|
| Base | Cilindro | X: 0.5m; Z: 0.08m |
| Cuerpo1 | Cilindro | X: 0.3m; Z: 0.2m |
| Cuerpo2 | Cilindro | X: 0.2m; Z: 0.22m |
| Hombro | Cilindro | X: 0.1m; Z: 0.2m |
| Brazo | Cuboide [Ortoedro] | X: 0.1m; Y: 0.1m; Z: 0.4m |
| Codo | Cilindro | X: 0.2m; Z: 0.1m |
| Antebrazo | Cuboide [Ortoedro] | X: 0.1m; Y: 0.1m; Z: 0.3m |
| Muñeca | Cuboide [Ortoedro] | X: 0.1m; Y: 0.1m; Z: 0.2m |
| Esfera | Esfera | X: 0.1m |

***Nota:*** *Los parámetros definidos en esta tabla corresponden a sus dimensiones en el mundo en el momento de su creación; posteriormente se han realizado las correspondientes traslaciones y rotaciones para ubicarlos en su posición de destino.*

Estos componentes configuran un robot con los siguientes parámetros espaciales (excluyendo de los cálculos la esfera final utilizada para localizar la punta):

a. Altura del cuerpo ($H_c$): 0.08m + 0.2m + 0.22m = 0.5 metros.
b. Longitud del brazo extendido ($L_b$): 0.2m + 0.4m + 0.1m + 0.3m + 0.2m = 1.2 metros.
c. Altura en estado inicial ($H_r$): 0.5m + ½ (0.1m) + ¼ (0.2m) = 0.6 metros.
d. Altura máxima alcanzable ($H_{R} = H_{c} + L_b$): 0.5m + 1.2m = 1.7 metros.
e. Longitud horizontal en estado inicial ($L_r$): 1.2m + ½ (0.2m) + ½ (0.5m) = 1.55 metros.
f. Longitud desde el punto central ($L_{r0}$): 1.2m + ½ (0.2m) = 1.3m
g. Longitud horizontal máxima ($L_R$): 1.55 metros.

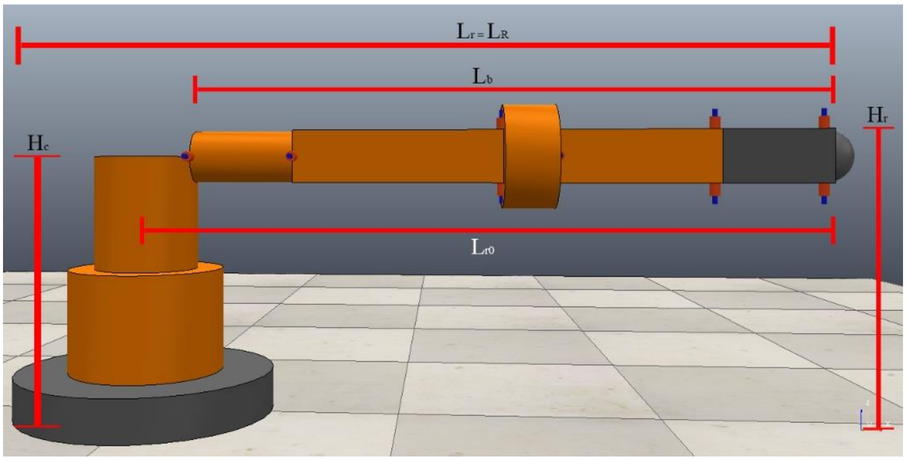


*Ilustración 1. Especificación de las medidas de nuestro robot.*

**Relación de articulaciones:**

La configuración de las diferentes articulaciones se ha realizado atendiendo a los parámetros de Denavit-Hartenberg, según se especifica paso por paso a continuación:

La articulación 1 (Art_1) es una articulación completamente vertical en el espacio tridimensional del propio mundo a una distancia del suelo de 0.08 metros -sobre la base negra-, mientras que la articulación 2 (Art_2) se ubicará de forma perpendicular en el eje Y y tendrá un desplazamiento de 0.1m sobre el eje X. De esta manera, siguiendo las transformaciones de Denavit-Hartenberg, obtenemos:

a. Giro en el eje Z un ángulo theta: no procede.
b. Desplazamiento en el eje Z una distancia d: desplazamiento de 0.42m para levantarla del suelo del mundo (esto permite ubicar la articulación en la parte superior del cuerpo: 0.08 + 0.42 = 0.5m en el eje vertical).
c. Giro en el eje X un ángulo alpha: hacemos un giro de 90 grados para colocar la nueva articulación perpendicularmente a la articulación 1.
d. Desplazamiento en el eje X una distancia a: realizamos un desplazamiento final de 0.1m (½ * X_Cuerpo2 = ½ * 0.2m) para posicionar la nueva articulación en el extremo de la parte superior del Cuerpo2.

```
Between 'Art_1' and 'Art_2':
    d=0.4200
    theta=-0.0
    a=0.1000
    alpha=90.0
```

La articulación 3 es paralela a la articulación 2, por lo que únicamente se realiza un desplazamiento en el eje X del nuevo sistema de coordenadas proporcionado por la articulación 2:

a. Giro en el eje Z un ángulo theta: no procede.
b. Desplazamiento en el eje Z una distancia d: no procede.
c. Giro en el eje X un ángulo alpha: no procede.
d. Desplazamiento en el eje X una distancia a: realizamos un desplazamiento de 0.2m, igual a la longitud del Hombro (Z_Hombro: 0.2m).

```
Between 'Art_2' and 'Art_3':
    d=0.0000
    theta=0.0
    a=0.2000
    alpha=0.0
```

Para calcular la ubicación de la articulación 4 respecto de la anterior, nos damos cuenta rápidamente que simplemente es preciso ejecutar una rotación y desplazamiento a lo largo del eje X.

a. Giro en el eje Z un ángulo theta: no procede.
b. Desplazamiento en el eje Z una distancia d: no procede.
c. Giro en el eje X un ángulo alpha: realizamos un giro de 90 grados en el eje X del nuevo sistema de coordenadas definido por la articulación 3 (igual al de la articulación 2 por ser articulaciones paralelas) para ubicar la nueva articulación perpendicularmente a su anterior.
d. Desplazamiento en el eje X una distancia a: realizamos un desplazamiento de 0.4m en el nuevo eje X, igual a la longitud del brazo (Z_Brazo: 0.4m).

```
Between 'Art_3' and 'Art_4':
    d=-0.0000
    theta=0.0
    a=0.4000
    alpha=90.0
```

En cuanto a la articulación 5, su cálculo es conceptualmente más sencillo porque los ejes de la articulación 4 vuelven a ser los del mundo. Para obtenerla, simplemente se realiza un giro de 90 grados sobre X y nos desplazamos la distancia correspondiente sobre el propio eje X.

a. Giro en el eje Z un ángulo theta: no procede.
b. Desplazamiento en el eje Z una distancia d: no procede.
c. Giro en el eje X un ángulo alpha: giro de 90 grados.
d. Desplazamiento en el eje X una distancia a: nos desplazamos una distancia a=0.1 metros, que es la distancia correspondiente a la longitud del codo (Z_Codo = 0.1m).

```
Between 'Art_4' and 'Art_5':
    d=-0.0000
    theta=-0.0
    a=0.1000
    alpha=90.0
```

Por último, la articulación 6 es perpendicular a la articulación 5, por lo que nuevamente se requiere un desplazamiento y rotación ortogonal en el eje X del sistema de coordenadas definido por la articulación 5, que vuelve a coincidir con el sistema de coordenadas del mundo.

a. Giro en el eje Z un ángulo theta: no procede.
b. Desplazamiento en el eje Z una distancia d: no procede.
c. Giro en el eje X un ángulo alpha: 90 grados.
d. Desplazamiento en el eje X una distancia a: desplazamiento de 0.3 metros, longitud asignada al antebrazo (Z_Antebrazo = 0.3m).

```
Between 'Art_5' and 'Art_6':
    d=0.0000
    theta=0.0
    a=0.3000
    alpha=90.0
```

## 2. Codificación en Matlab del algoritmo de Denavit Hartenberg

Con los parámetros especificados, es momento de escribir nuestro Script de Matlab para el cálculo de la posición final del robot a partir de los parámetros de configuración de las respectivas articulaciones. En primer lugar, conviene especificar la estrategia seguida para el cálculo del algoritmo de Denavit-Hartenberg:

```matlab
function DH = DenavitHartenberg(d, theta, a, alpha)
theta = theta*pi/180;
alpha = alpha*pi/180;

Rz = [ cos(theta) -sin(theta) 0 0
       sin(theta)  cos(theta) 0 0
                0           0 1 0
                0           0 0 1  ];

Tz = [1 0 0 0
      0 1 0 0
      0 0 1 d
      0 0 0 1];

Tx = [1 0 0 a
      0 1 0 0
      0 0 1 0
      0 0 0 1];

Rx = [1          0           0      0
      0 cos(alpha) -sin(alpha)      0
      0 sin(alpha)  cos(alpha)      0
      0          0           0      1 ];


%La matriz DH_12 es la matriz de transformación de art_1 a art_2
DH = Rz * Tz * Tx * Rx;
end
```

En primer lugar, se define una función que nos calcula la matriz de paso de una articulación [i-1] a la siguiente [i], a partir de los parámetros de distancia (d, a) y ángulo de giro (theta, alpha) sobre los ejes Z y X, respectivamente, asignados a la articulación de partida. Como se ha visto en teoría, simplemente será necesario calcular ordenadamente las matrices de Rotación en Z (Rz), Traslación en Z (Tz), Traslación en X (Tx) y Rotación en X (Rx), y multiplicarlas finalmente para obtener la matriz de transformación a la nueva articulación.

Una vez tenemos esta función, simplemente nos iremos apoyando en ella con los valores correspondientes a cada articulación del robot para obtener los pasos de articulación necesarios. Sirva como ejemplo el siguiente fragmento de código para ilustrar el cálculo de la transformación de la articulación 1 a la 2.

```
desplazamiento = 0.08;
angulo = 0;
d = 0.42 + desplazamiento;
theta = 0 + angulo;
a = 0.1;
alpha = 90;

DH_12 = DenavitHartenberg(d, theta, a, alpha);
disp("Transformación de Denavit Hartenberg: Articulación 1 -> Articulación 2");
disp(DH_12);
```

Al final, todas estas matrices de paso se multiplican para obtener la matriz homogénea de Denavit Hartenberg, que nos proporciona la posición final del robot con los parámetros especificados.

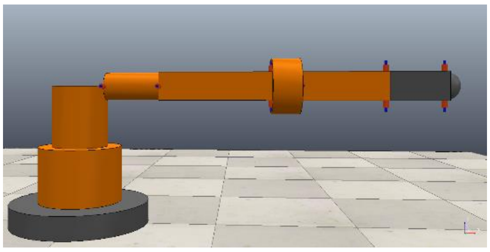
```
%Matriz de Transformación Homogénea Final (DH_F):
DH_F = DH_12 * DH_23 * DH_34 * DH_45 * DH_56 * DH_6F;

disp("Transformación Homogénea Final de Denavit Hartenberg:");
disp(DH_F);
```

## 3. Resultados experimentales con diferentes configuraciones

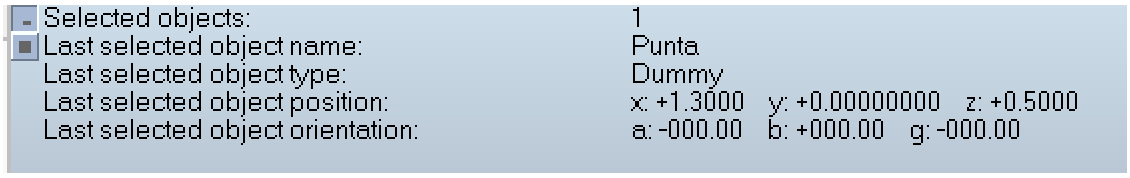
Se ha probado el sistema en tres configuraciones diferentes, variando los ángulos de rotación de las diferentes articulaciones según se explica en las tablas anexas:

### a. Configuración inicial del robot

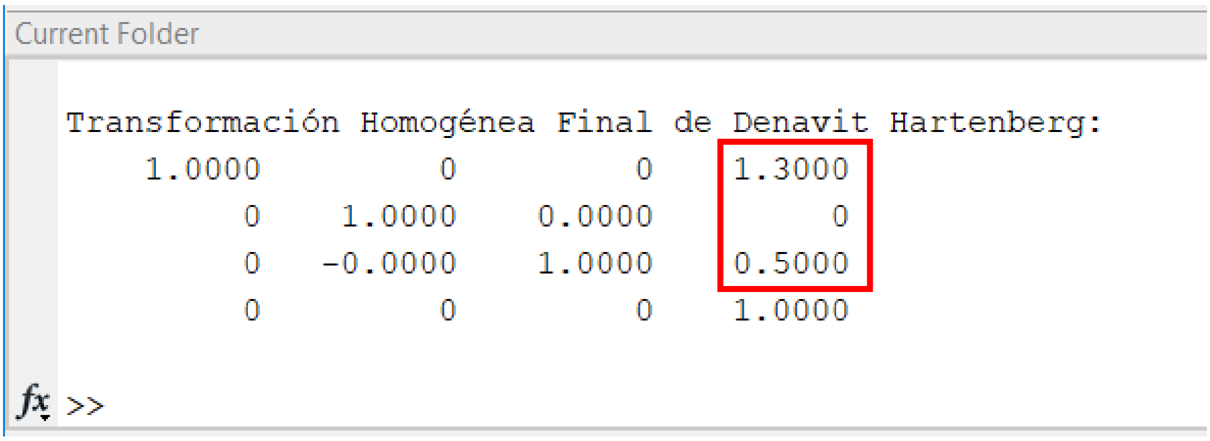
| Identificador de la Articulación | Ángulo de Giro |
|---|---|
| Art_1 | 0º |
| Art_2 | 0º |
| Art_3 | 0º |
| Art_4 | 0º |
| Art_5 | 0º |
| Art_6 | 0º |



Si nos apoyamos directamente en el propio simulador, CoppeliaSim y obtenemos la posición del elemento final, llamado Punta, nos ofrece los siguientes resultados:



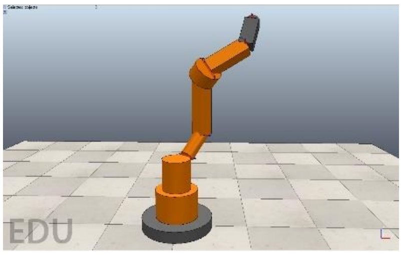

La imagen anterior nos permite definir el vector de salida del robot, es decir, el vector espacial que nos indica exactamente dónde está ubicada la Punta, esto es, (1.3, 0.0, 0.5). Acudimos a nuestro Script de Matlab para comprobar que, efectivamente, la matriz homogénea nos proporciona la misma información, como es lógico:




Precisamente, si nos fijamos en los parámetros señalados de la última columna de la matriz homogénea, obtenemos el mismo vector de salida: (1.3, 0.0, 0.5).

### b. Configuración con idénticos ángulos de rotación

| **Identificador de la Articulación** | **Ángulo de Giro** |
|---|---|
| Art_1 | 45º |
| Art_2 | 45º |
| Art_3 | 45º |
| Art_4 | 45º |
| Art_5 | 45º |
| Art_6 | 45º |




Para realizar un experimento intermedio, decidimos repetir el proceso con una configuración distinta de la posición inicial de robot, si bien aplicando en todas las articulaciones un mismo ángulo de rotación de 45 grados. Los resultados vuelven a ser coherentes, como a continuación se refleja.

Obtenemos los datos de posición de la Punta proporcionados por el simulador, que nos define como vector de salida el vector (0.5268, 0.2561, 1.4328):

```
Selected objects:                    1
Last selected object name:           Punta
Last selected object type:           Dummy
Last selected object position:       x: +0.5268   y: +0.2561   z: +1.4328
Last selected object orientation:    a: +059.64   b: -008.42   g: +075.36
```

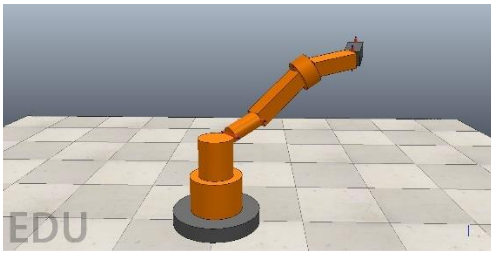
Realizamos la misma comprobación en nuestro Script de Matlab, añadiendo un ángulo de rotación de 45 grados al parámetro theta de cada articulación:

```
Current Folder

Transformación Homogénea Final de Denavit Hartenberg:
    0.2500   -0.9571   -0.1464    0.5268
    0.4571    0.2500   -0.8536    0.2561
    0.8536    0.1464    0.5000    1.4328
         0         0         0    1.0000

fx >>
```

Nuevamente, se comprueba que el vector de salida coincide tras ejecutar el Script.

### c. Configuración con rotaciones aleatorias

| Identificador de la Articulación | Ángulo de Giro |
|---|---|
| Art_1 | 38º |
| Art_2 | 20º |
| Art_3 | 14º |
| Art_4 | 19º |
| Art_5 | 23º |
| Art_6 | 51º |




Una vez realizadas las comprobaciones más elementales, debemos probar la consistencia entre nuestro modelo y el Script con datos de configuración totalmente arbitrarios. Los ángulos con que se ha configurado cada articulación con los referidos en la tabla anterior. Una vez más, comenzamos por conocer el vector de salida en CoppeliaSim, que nos devuelve la posición (0.9067, 0.6916, 0.9426).

```
Selected objects:                    1
Last selected object name:           Punta
Last selected object type:           Dummy
Last selected object position:       x: +0.9067   y: +0.6916   z: +0.9426
Last selected object orientation:    a: +013.26   b: -004.93   g: +072.77
```

Acudimos al Script con estos parámetros para verificar que, efectivamente, el vector de salida dado por el simulador y el vector extraído de la matriz homogénea de Denavit Hartenberg, coinciden:

```
Current Folder

Transformación Homogénea Final de Denavit Hartenberg:
    0.2952   -0.9516   -0.0859    0.9067
    0.9238    0.3072   -0.2286    0.6916
    0.2439   -0.0119    0.9697    0.9426
         0         0         0    1.0000

fx >>
```