

UNIVERSIDAD POLITÉCNICA DE MADRID

ROBÓTICA
MANIPULADORES

Estudio del manipulador μArm

Autores:

Javier Alonso Silva - javier.asilva@alumnos.upm.es

Roberto Álvarez Garrido - roberto.alvarezg@alumnos.upm.es

José Alejandro Moya Blanco - alejandro.moya.blanco@alumnos.upm.es

Última modificación: 15 de noviembre de 2019



Conocimientos previos

Antes de ponernos a hablar sobre los resultados obtenidos en la práctica, antes vamos a hablar sobre algunas características básicas del brazo robótico e introducirlo brevemente. El manipulador robótico μArm es un dispositivo creado por la empresa [UFACTORY](#) el cual cuenta con cuatro grados de libertad. De dichos grados de libertad, tres son usados para mover el brazo robótico hasta ciertas posiciones y, el último, para mantener el extremo del mismo paralelo al suelo.

El manipulador es controlado mediante cuatro motores:

El **motor de la base** el cual permite la rotación del manipulador.

En el brazo, el **motor que está a la derecha** (ver la figura 1), coordina el movimiento de la parte inferior del brazo (*Lower Arm* en la figura) con la parte superior del mismo (*Upper Arm* en la figura).

En esta parte del manipulador, el movimiento es como el de un flexo: la parte superior del flexo está supeditada a la parte inferior, de manera que se mantiene de forma constante la altura a la que está el extremo final del mismo.

El otro motor, localizado a la **izquierda del brazo**, se encarga de mantener la orientación del extremo del manipulador. De esta manera, dicho extremo permanecerá paralelo al suelo. Teniendo en cuenta esto, podríamos decir que el robot en verdad solo tiene tres grados de libertad en tanto a que no se controla directamente el movimiento del último grado, ya que al final se mueve para permanecer paralelo al suelo.

El **motor localizado en el extremo**, con el cual se puede actuar sobre el elemento que esté colocado allí. Por ejemplo, cuando se coloca una ventosa permite rotarla o, cuando se coloca la pinza, el movimiento del motor permite abrirla o cerrarla.

Para este estudio, este último motor se descartará, ya que no afecta a las posiciones accesibles por el robot.

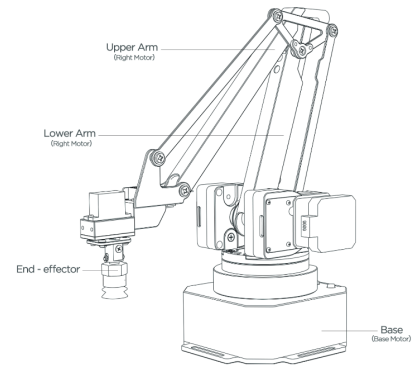


Figura 1: zonas de actuación de los motores en el brazo [1]

<i>Motor</i>	<i>Rango de trabajo</i>
Base	$0^\circ \sim 180^\circ$
Derecho	$0^\circ \sim 130^\circ$
Izquierdo	$0^\circ \sim 106^\circ$
Extremo	$0^\circ \sim 180^\circ$

Cuadro 1: ángulo de giro de los motores

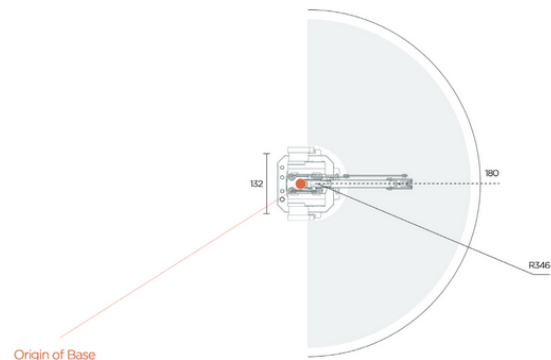


Figura 2: rango del manipulador μArm [2]

Toda la información relativa al desarrollo del proyecto puede ser encontrada en [GitHub - UPM Robotics \[3\]](#). Allí están detallados los distintos hitos a conseguir así como más información sobre el robot.

Además, se encuentra disponible la siguiente bibliografía:

- [Manual de usuario](#)
- [Especificaciones](#)
- [Guía del desarrollador](#)
- [Modelo en 3D](#)
- [Web de UFACTORY](#)
- [Soporte de UFACTORY](#)

1. Configuración geométrica

En esta sección vamos a describir la configuración geométrica del brazo robótico. La configuración que obtuvimos fue la siguiente:

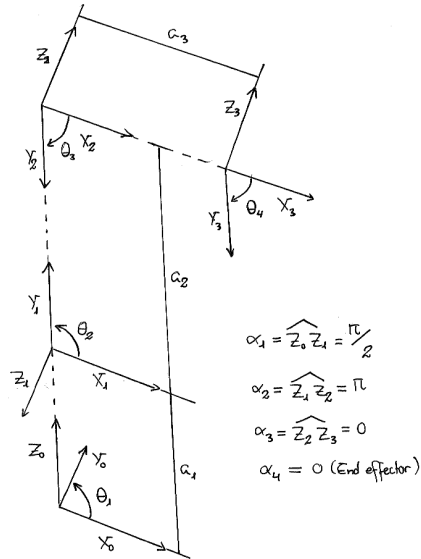


Figura 3: configuración geométrica del robot

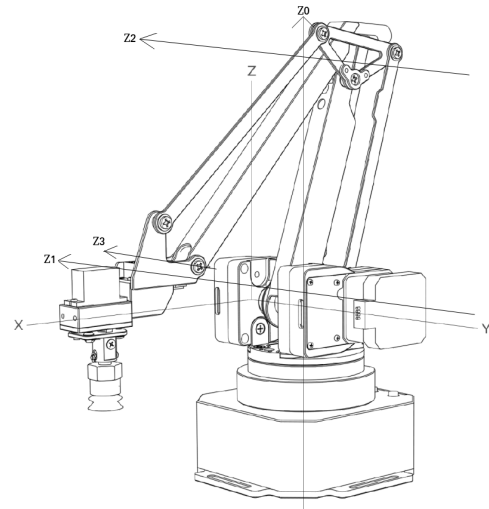


Figura 4: los grados de libertad del brazo, representados por los diferentes Z_i

Usando los datos que están presentes en la documentación al desarrollador [1], pudimos obtener los siguientes datos para los a_i (distancia entre ejes) del manipulador; además, descubrimos que hay una pequeña desviación d_i entre las articulaciones $\{1, 2\}$:

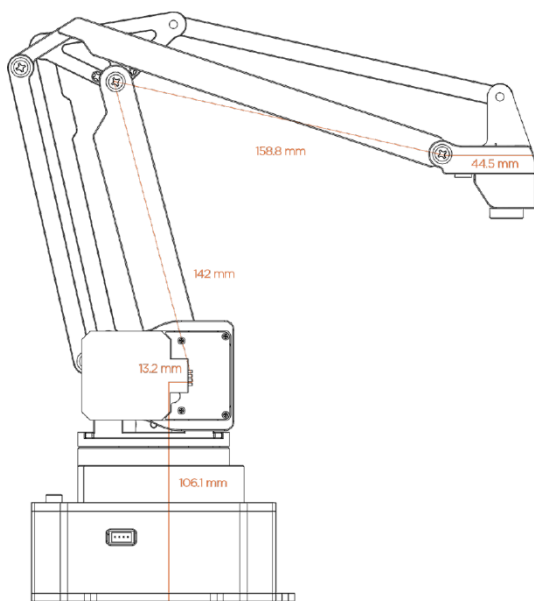


Figura 5: longitudes del brazo robótico [1]

i	a_i (mm.)	d_i (mm.)
1	106,1	13,2
2	142	0
3	158,8	0
4	44,5	0

Cuadro 2: longitudes y desviaciones del manipulador

De esta manera, con los datos obtenidos, generamos una primera matriz de *Denavit-Hartenberg*:

i	θ_i	d_i (mm.)	a_i (mm.)	α_i
1	θ_1	13,2	106,1	$\frac{\pi}{2}$
2	θ_2	0	142	π
3	θ_3	0	158,8	0
4	θ_4	0	44,5	0

Cuadro 3: primera tabla de *Denavit-Hartenberg*

i	θ_i	d_i (mm.)	a_i (mm.)	α_i
1	θ_1	d_1	a_1	$\frac{\pi}{2}$
2	θ_2	0	a_2	π
3	θ_3	0	a_3	0
4	θ_4	0	a_4	0

Cuadro 4: primera tabla de *Denavit-Hartenberg* parametrizada

La cuestión es que, al hacer los distintos modelos, descubrimos que debido a la orientación del brazo robótico, la d_i está en la posición del equivalente a_i , en particular d_1 y a_1 . Esto es debido a que, como se puede ver en la figura 4 junto con las figuras 3 y 5, el plano sobre el que está d_i es el XZ , lo cual nos obliga a cambiar las posiciones en las que existen a la vez un a_i y un d_i , siempre y cuando ambos sean constantes, que en este caso lo son. De esta forma, la tabla de *Denavit-Hartenberg* quedaría:

i	θ_i	d_i (mm.)	a_i (mm.)	α_i
1	θ_1	106,1	13,2	$\frac{\pi}{2}$
2	θ_2	0	142	π
3	θ_3	0	158,8	0
4	θ_4	0	44,5	0

Cuadro 5: segunda tabla de *Denavit-Hartenberg*

i	θ_i	d_i (mm.)	a_i (mm.)	α_i
1	θ_1	a_1	d_1	$\frac{\pi}{2}$
2	θ_2	0	a_2	π
3	θ_3	0	a_3	0
4	θ_4	0	a_4	0

Cuadro 6: segunda tabla de *Denavit-Hartenberg* parametrizada

Finalmente, el *end-effector* (véase la figura 1) siempre está perpendicular al plano del suelo, es decir, $\phi_e = \pi$. Por eso mismo, el parámetro $i = 4$ en verdad está supeditado siempre al movimiento que se realice según los ángulos θ_2 y θ_3 , así que no es necesario contemplarlo en la tabla de *Denavit-Hartenberg*:

i	θ_i	d_i (mm.)	a_i (mm.)	α_i
1	θ_1	106,1	13,2	$\frac{\pi}{2}$
2	θ_2	0	142	π
3	θ_3	0	158,8	0

Cuadro 7: tabla de *Denavit-Hartenberg*

i	θ_i	d_i (mm.)	a_i (mm.)	α_i
1	θ_1	a_1	d_1	$\frac{\pi}{2}$
2	θ_2	0	a_2	π
3	θ_3	0	a_3	0

Cuadro 8: tabla de *Denavit-Hartenberg* parametrizada

2. Cinemática directa

A continuación, con los valores obtenidos en el apartado anterior, vamos a obtener las distintas matrices de transformación de referenciales de manipuladores. Para ello, partimos de la siguiente matriz:

$$A_{i-1}^i = \begin{pmatrix} \cos \theta_i & -\cos \alpha_i \sin \theta_i & \sin \alpha_i \sin \theta_i & a_i \cos \theta_i \\ \sin \theta_i & \cos \alpha_i \cos \theta_i & -\sin \alpha_i \cos \theta_i & a_i \sin \theta_i \\ 0 & \sin \alpha_i & \cos \alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Aplicando los distintos pasos, obtenemos:

$$A_0^1 = \begin{pmatrix} \cos(\theta_1) & 0 & \sin(\theta_1) & d_1 \cos(\theta_1) \\ \sin(\theta_1) & 0 & -\cos(\theta_1) & d_1 \sin(\theta_1) \\ 0 & 1 & 0 & a_1 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$A_1^2 = \begin{pmatrix} \cos(\theta_2) & \sin(\theta_2) & 0 & a_2 \cos(\theta_2) \\ \sin(\theta_2) & -\cos(\theta_2) & 0 & a_2 \sin(\theta_2) \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$A_2^3 = \begin{pmatrix} \cos(\theta_3) & -\sin(\theta_3) & 0 & a_3 \cos(\theta_3) \\ \sin(\theta_3) & \cos(\theta_3) & 0 & a_3 \sin(\theta_3) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Con todas las matrices ya obtenidas, podemos calcular la matriz de transformación directa del manipulador (son 4 columnas, pero no cabe la matriz al completo, por eso la última columna está debajo):

$$A_0^2 = \begin{pmatrix} \cos(\theta_1) \cos(\theta_2) & \sin(\theta_2) \cos(\theta_1) & -\sin(\theta_1) & (a_2 \cos(\theta_2) + d_1) \cos(\theta_1) \\ \sin(\theta_1) \cos(\theta_2) & \sin(\theta_1) \sin(\theta_2) & \cos(\theta_1) & (a_2 \cos(\theta_2) + d_1) \sin(\theta_1) \\ \sin(\theta_2) & -\cos(\theta_2) & 0 & a_1 + a_2 \sin(\theta_2) \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$A_0^3 = \begin{pmatrix} \cos(\theta_1) \cos(\theta_2 - \theta_3) & \sin(\theta_2 - \theta_3) \cos(\theta_1) & -\sin(\theta_1) \\ \sin(\theta_1) \cos(\theta_2 - \theta_3) & \sin(\theta_1) \sin(\theta_2 - \theta_3) & \cos(\theta_1) \\ \sin(\theta_2 - \theta_3) & -\cos(\theta_2 - \theta_3) & 0 \\ 0 & 0 & 0 \\ (a_2 \cos(\theta_2) + a_3 \cos(\theta_2 - \theta_3) + d_1) \cos(\theta_1) & (a_2 \cos(\theta_2) + a_3 \cos(\theta_2 - \theta_3) + d_1) \sin(\theta_1) & a_1 + a_2 \sin(\theta_2) + a_3 \sin(\theta_2 - \theta_3) \\ & & 1 \end{pmatrix}$$

Las matrices están parametrizadas. Numéricamente, la matriz de transformación directa A_0^3 quedaría:

$$A_0^3 = \begin{pmatrix} \cos(\theta_1) \cos(\theta_2 - \theta_3) & \sin(\theta_2 - \theta_3) \cos(\theta_1) & -\sin(\theta_1) \\ \sin(\theta_1) \cos(\theta_2 - \theta_3) & \sin(\theta_1) \sin(\theta_2 - \theta_3) & \cos(\theta_1) \\ \sin(\theta_2 - \theta_3) & -\cos(\theta_2 - \theta_3) & 0 \\ 0 & 0 & 0 \\ (142,0 \cos(\theta_2) + 158,9 \cos(\theta_2 - \theta_3) + 13,2) \cos(\theta_1) & (142,0 \cos(\theta_2) + 158,9 \cos(\theta_2 - \theta_3) + 13,2) \sin(\theta_1) & 142,0 \sin(\theta_2) + 158,9 \sin(\theta_2 - \theta_3) + 106,1 \\ & & 1 \end{pmatrix}$$

Finalmente, hay que añadir una traslación en el eje Z (debido a la posición del *end-effector*) y en el eje X , ya que después de la articulación θ_3 hay una extensión de $44,5 \text{ mm}$. (ver figura 5), para obtener así la posición final del robot ($X_e Y_e Z_e$):

$$A_0^3 = \begin{pmatrix} \cos(\theta_1) \cos(\theta_2 - \theta_3) & \sin(\theta_2 - \theta_3) \cos(\theta_1) & -\sin(\theta_1) \\ \sin(\theta_1) \cos(\theta_2 - \theta_3) & \sin(\theta_1) \sin(\theta_2 - \theta_3) & \cos(\theta_1) \\ \sin(\theta_2 - \theta_3) & -\cos(\theta_2 - \theta_3) & 0 \\ 0 & 0 & 0 \\ (a_2 \cos(\theta_2) + a_3 \cos(\theta_2 - \theta_3) + d_1) \cos(\theta_1) + T_X & (a_2 \cos(\theta_2) + a_3 \cos(\theta_2 - \theta_3) + d_1) \sin(\theta_1) & a_1 + a_2 \sin(\theta_2) + a_3 \sin(\theta_2 - \theta_3) - T_Z \\ 1 & & \end{pmatrix}$$

$$A_0^3 = \begin{pmatrix} \cos(\theta_1) \cos(\theta_2 - \theta_3) & \sin(\theta_2 - \theta_3) \cos(\theta_1) & -\sin(\theta_1) \\ \sin(\theta_1) \cos(\theta_2 - \theta_3) & \sin(\theta_1) \sin(\theta_2 - \theta_3) & \cos(\theta_1) \\ \sin(\theta_2 - \theta_3) & -\cos(\theta_2 - \theta_3) & 0 \\ 0 & 0 & 0 \\ (142,0 \cos(\theta_2) + 158,9 \cos(\theta_2 - \theta_3) + 13,2) \cos(\theta_1) + 44,5 & (142,0 \cos(\theta_2) + 158,9 \cos(\theta_2 - \theta_3) + 13,2) \sin(\theta_1) & 142,0 \sin(\theta_2) + 158,9 \sin(\theta_2 - \theta_3) + 106,1 - T_Z \\ 1 & & \end{pmatrix}$$

Referencias

- [1] *uArm Swift Pro_Developer Guide v1.0.6.pdf*, en, 2019. dirección: http://download.ufactory.cc/docs/en/uArm%20Swift%20Pro_Developer%20Guide%20v1.0.6.pdf (visitado 02-11-2019).
- [2] *uArm pro User Manual v1.1.0.pdf*, en, 2019. dirección: <http://download.ufactory.cc/docs/en/uArm%20pro%20User%20Manual%20v1.1.0.pdf> (visitado 02-11-2019).
- [3] *UPM-Robotics/uarm*, original-date: 2019-11-01T11:13:54Z, nov. de 2019. dirección: <https://github.com/UPM-Robotics/uarm> (visitado 02-11-2019).