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: 16 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.

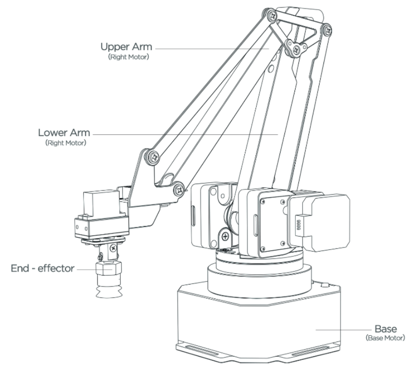

Figura 1: zonas de actuación de los motores en el brazo [1]

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.

Motor	*Rango de trabajo*
Base	0° ∼ 180°
Derecho	0° ∼ 130°
Izquierdo	0° ∼ 106°
Extremo	0° ∼ 180°

Cuadro 1: ángulo de giro de los motores

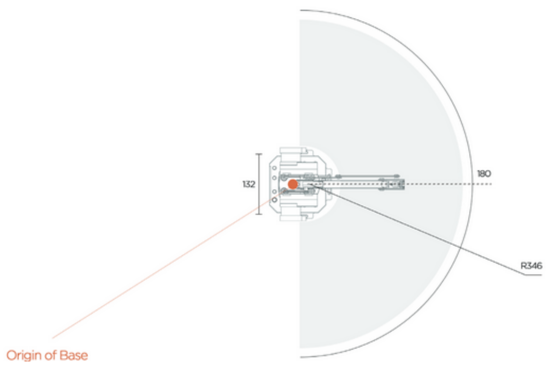

Figura 2: rango del manipulador μArm [2]

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:

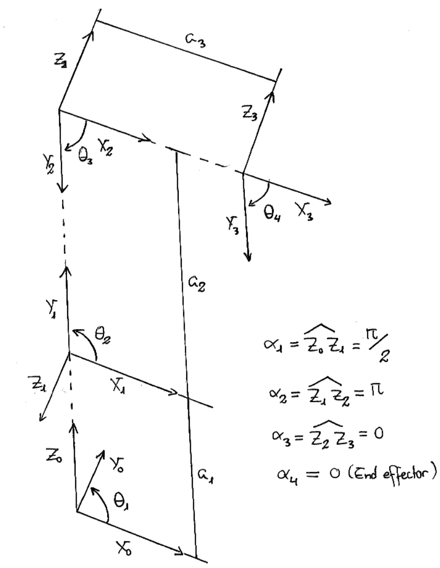

Figura 3: configuración geométrica del robot

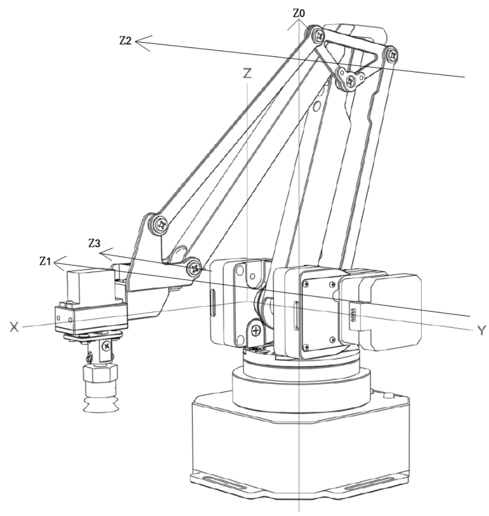

Figura 4: los grados de libertar del brazo, representados por los diferentes Z_i

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 $\{1, 2\}$:

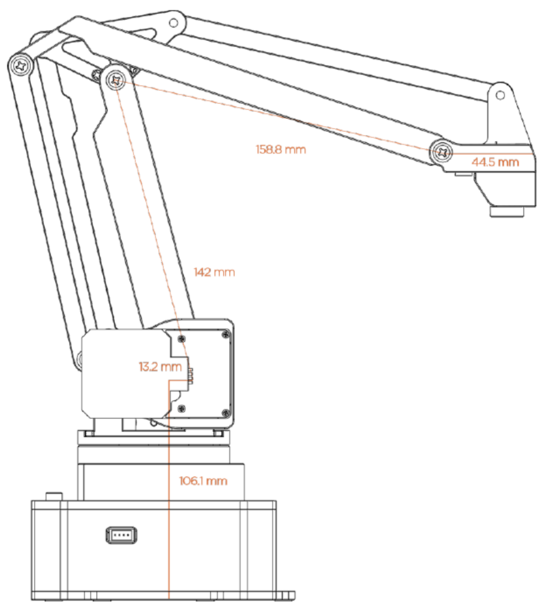

Figura 5: longitudes del brazo robótico [1]

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, sin embargo no cabe la matriz al completo y 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[1] 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 $44,5$ $mm.$ (ver figura 5), para obtener así la posición final del robot (X_e Y_e Z_e):

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

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

De esta forma, obtendríamos las siguientes ecuaciones para (X_e Y_e Z_e):

$$\left.\begin{aligned} X_e &= (a_2\cos(\theta_2)+a_3\cos(\theta_2-\theta_3)+d_1)\cos(\theta_1)+T_X \\ Y_e &= (a_2\cos(\theta_2)+a_3\cos(\theta_2-\theta_3)+d_1)\sin(\theta_1) \\ Z_e &= a_1+a_2\sin(\theta_2)+a_3\sin(\theta_2-\theta_3)-T_Z \end{aligned}\right\} \tag{3}$$

[1] T_X es la traslación en X, mientras que T_Z es la traslación en Z

3. Cinemática inversa

Una vez tenemos las ecuaciones correspondientes a $(X_e \; Y_e \; Z_e)$, podemos calcular la cinemática inversa del μArm. Con dicha cinemática, se pueden obtener los ángulos $(\theta_1, \theta_2, \theta_3)$ del brazo robótico.

Antes de realizar los cálculos, se planteó el modelo geométrico del robot, para poder deducir los distintos ángulos que se podían formar y cómo estaba distribuido el brazo:

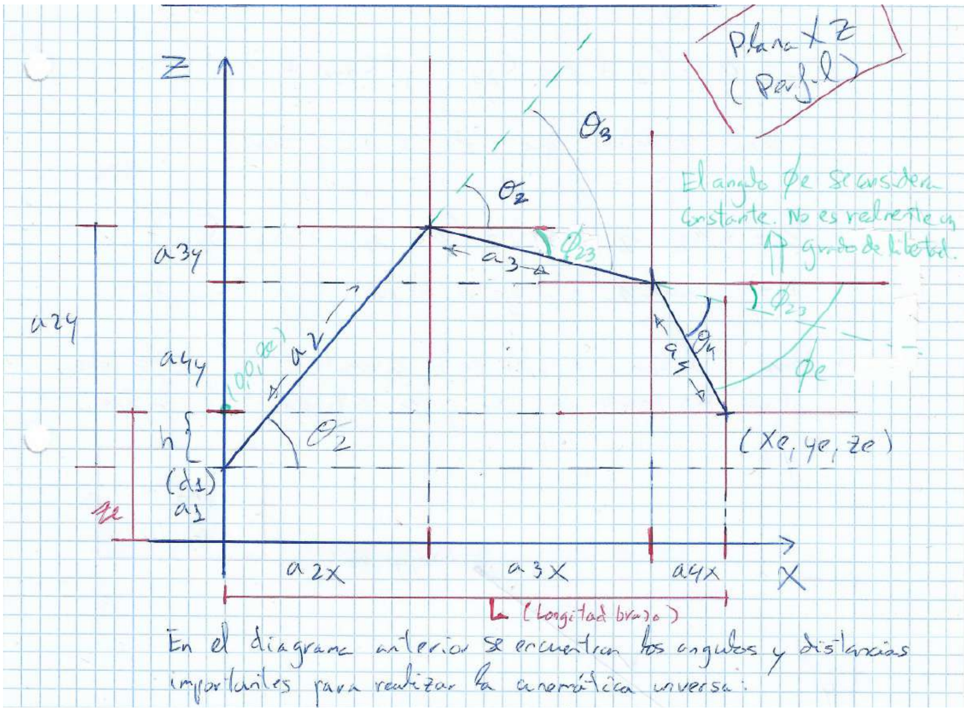

Figura 6: distribución geométrica del brazo

En la figura 6 se puede ver cómo, en el plano XZ (de perfil), se encuentra presente una altura inicial a_1 correspondiente a la base del brazo, así como una pequeña desviación en los ejes d_1 la cual afecta a la posición en Z. A continuación, cada extremidad del brazo está definida junto con la correspondiente articulación θ_i. Como se puede observar, el ángulo θ_2 es positivo pero, debido a cómo está configurado el robot geométricamente, los ángulos θ_3 y θ_4 han de ser negativos[2].

Como se vio anteriormente en la sección *Conocimientos previos*, en particular la figura 3, se suprimió el último parámetro de la tabla de *Denavit–Hartenberg* ya que dicho ángulo siempre es perpendicular al plano del suelo y, por ende, está supeditado a los ángulos anteriores θ_2 y θ_3. En su lugar, se sustituye por una traslación en el eje Z: T_Z (véase las matrices 1 y 2 y la ecuación 3).

De esta forma, se obtuvieron las siguientes ecuaciones para el plano XZ:

$$L = \begin{cases} a_{2X} = a_2 \cdot \cos(\theta_2) \\ a_{3X} = a_3 \cdot \cos(\theta_2 + \theta_3) \\ a_{4X} = a_4 \cdot \cos(\phi_e) \end{cases} \tag{4}$$

$$H = \begin{cases} a_{2Z} = a_2 \cdot \sin(\theta_2) \\ a_{3Z} = a_3 \cdot \sin(\theta_2 + \theta_3) \\ a_{4Z} = a_4 \cdot \sin(\phi_e) \end{cases} \tag{5}$$

[2] véase la figura 5 para más información sobre la configuración del brazo robótico

Finalmente, se evaluó el plano XY del modelo geométrico del brazo robótico:

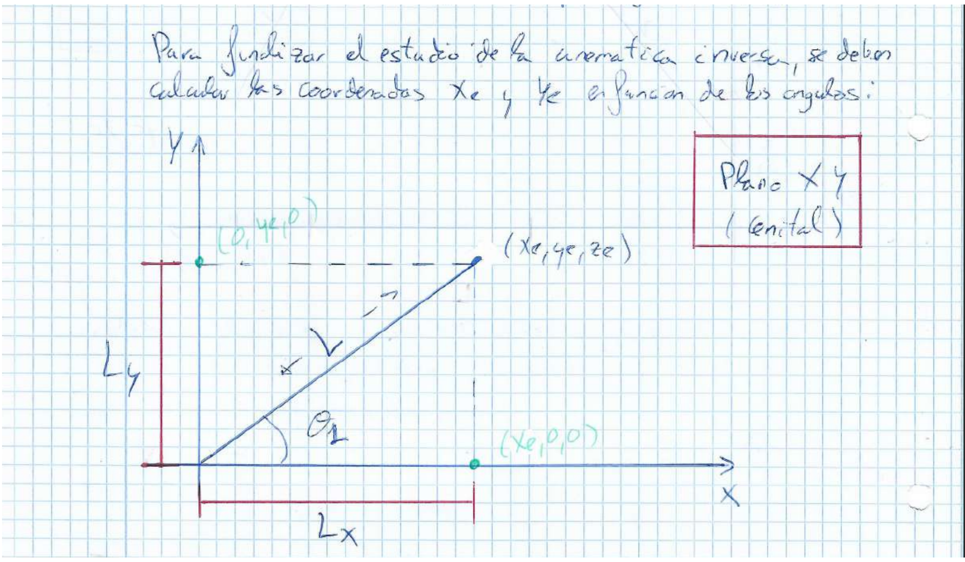

Figura 7: plano XY del brazo robótico

De esta manera, se obtuvieron las siguientes ecuaciones para L_X y L_Y:

$$\begin{cases} L_X = L \cdot \cos(\theta_1) = X_e \\ L_Y = L \cdot \sin(\theta_1) = Y_e \end{cases} \tag{6}$$

La problemática de plantear el modelo de esta forma se mostró a la hora de intentar resolver las ecuaciones de X_e, Y_e y Z_e: al sustituir en la ecuación 3, el sistema a resolver era demasiado complejo, resultando imposible la obtención de alguno de los ángulos, con lo que el sistema no tenía solución.

Como ya se comentó anteriormente, en la tabla de *Denavit-Hartenberg* se eliminó el último parámetro: pese a que sea un grado de libertad, al estar supeditado a los dos ángulos anteriores y ser siempre perpendicular al plano del suelo, no es necesario tenerlo en cuenta para la obtención de la inversa del robot.

Esto produjo que el último punto efectivo del robot resultara ser el extremo del brazo de la articulación θ_3:

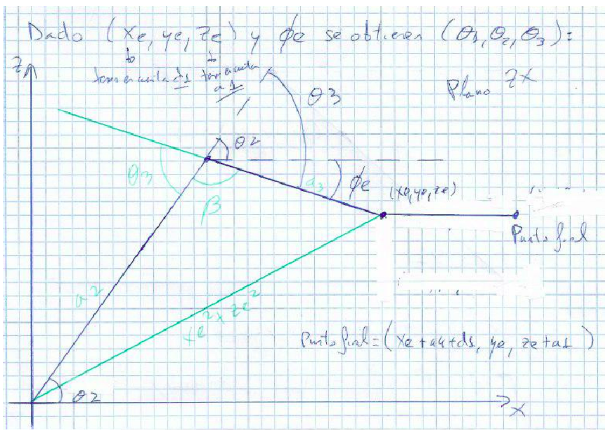

Figura 8: aproximación geométrica del μArm

De esta forma, la complejidad se reduce bastante, ya que ahora en el plano XZ solo se cuentan con dos grados de libertad: θ_2 y θ_3, además de la traslación en X que se comentó anteriormente (T_X) y que se puede ver en la figura 8.

Esta nueva configuración permite aplicar el teorema del coseno[3]:

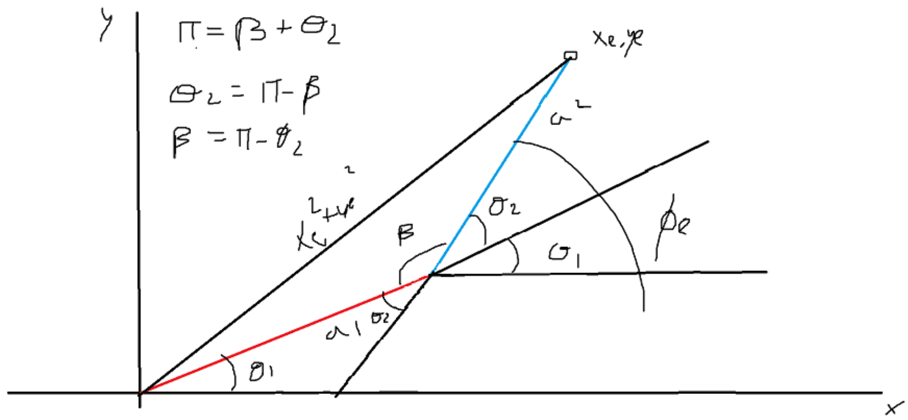

Figura 9: configuración del brazo robótico donde se puede aplicar el teorema del coseno

Así, se pueden extraer θ_2 y θ_3:

$$\pi = \beta + \theta_3 \Longrightarrow \beta = \pi - \theta_3 \tag{7}$$
$$\phi_e = \theta_2 - \theta_3 \tag{8}$$

Usando la ecuación 3, podemos despejar:

$$\begin{aligned} X_e^2 + Z_e^2 &= a_2^2 + a_3^2 - 2a_2a_3\cos(\beta) = \\ &= a_2^2 + a_3^2 - 2a_2a_3\cos(\pi - \theta_3) = \\ &= a_2^2 + a_3^2 + 2a_2a_3\cos(\theta_3) \end{aligned}$$

$$\cos(\theta_3) = \frac{X_e^2 + Y_e^2 - a_2^2 - a_3^2}{2a_2a_3} \tag{9}$$
$$\sin(\theta_3) = \sqrt{1 - \cos^2(\theta_3)} \tag{10}$$
$$\Longrightarrow \theta_3 = \arctan\left(\frac{\sin(\theta_3)}{\cos(\theta_3)}\right) \tag{11}$$
$$\Longrightarrow \theta_2 = \phi_e + \theta_3 \tag{12}$$
$$\tag{13}$$

Finalmente, en el plano XY tenemos la siguiente configuración:

[3]Teorema del coseno: $c^2 = a^2 + b^2 - 2ab\cos\beta$

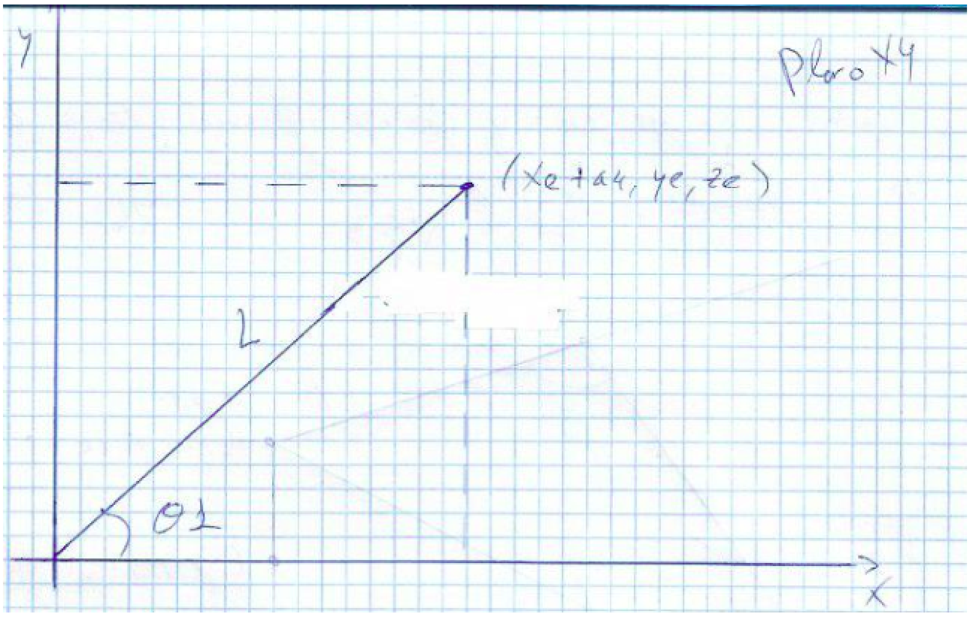

Figura 10: visión del μArm en el plano XY

De este modo, aplicando el teorema de Pitágoras[4], se obtiene:

$$L = \pm\sqrt{(X_e + T_X)^2 + Y_e^2}$$
$$\cos(\theta_1) = \frac{X_e + T_X}{\sqrt{(X_e + T_X)^2 + Y_e^2 + d_1}}$$
$$\sin(\theta_1) = \frac{Y_e}{\sqrt{(X_e + T_X)^2 + Y_e + d_1}}$$
$$\Longrightarrow \tan(\theta_1) = \frac{\sin(\theta_1)}{\cos(\theta_1)}$$

$$\theta_1 = \arctan\left(\frac{\frac{Y_e}{\sqrt{(X_e + T_X)^2 + Y_e^2 + d_1}}}{\frac{X_e + T_X}{\sqrt{(X_e + T_X)^2 + Y_e^2 + d_1}}}\right) \tag{14}$$
$$= \arctan\left(\frac{Y_e}{X_e + T_X + d_1}\right) \tag{15}$$
$$\tag{16}$$

De esta forma, un algoritmo de obtención de la inversa sería:

- Obtener θ_1 a partir del punto $(X_e + T_X, Y_e, Z_e - T_Z)$, usando la ecuación 15.
- Obtener $(X_e\ Y_e\ Z_e)$ quitando las traslaciones T_X y T_Z: $X_e = X_e - T_X$; $Z_e = Z_e + T_Z$.
- Obtener $\cos(\theta_3)$ con la ecuación 9 y el $\sin(\theta_3)$ con la ecuación 10.

[4]Teorema de Pitágoras: $h^2 = c_1^2 + c_2^2$

- Calcular θ_3 usando la ecuación 11.
- Calcular θ_2 usando la ecuación 12.

4. Matriz Jacobiana

La matriz Jacobiana nos permite obtener el movimiento $\vec{x}$, correspondiente a las velocidades en el extremo del robot dadas unas velocidades de las articulaciones $\vec{q}$.

La matriz Jacobiana es una matriz de derivadas, esto es, varía según el tiempo. De esta forma, se puede definir la matriz Jacobiana de un manipulador como:

$$\vec{x} = J \cdot \vec{q} \begin{cases} \vec{q} = (\theta_1, \cdots, \theta_n, d_1, \cdots, d_n) \\ \vec{x} = (X_e, Y_e, Z_e, \phi_e) \\ J \equiv matriz\ Jacobiana \end{cases} \tag{17}$$

$$J_1(\vec{q}) = \begin{pmatrix} \frac{\partial X_e}{\partial q_0} & \frac{\partial X_e}{q_1} & \dots & \frac{\partial X_e}{q_n} & \frac{\partial X_e}{d_1} & \dots & \frac{\partial X_e}{d_n} \\ \frac{\partial Y_e}{\partial q_0} & \frac{\partial Y_e}{q_1} & \dots & \frac{\partial Y_e}{q_n} & \frac{\partial Y_e}{d_1} & \dots & \frac{\partial Y_e}{d_n} \\ \frac{\partial Z_e}{\partial q_0} & \frac{\partial Z_e}{q_1} & \dots & \frac{\partial Z_e}{q_n} & \frac{\partial Z_e}{d_1} & \dots & \frac{\partial Z_e}{d_n} \end{pmatrix} \tag{18}$$

$$J_2(\vec{q}) = \begin{pmatrix} \frac{\partial \phi_X}{\partial q_0} & \frac{\partial \phi_X}{q_1} & \dots & \frac{\partial \phi_X}{q_n} & \frac{\partial \phi_X}{d_1} & \dots & \frac{\partial \phi_X}{d_n} \\ \frac{\partial \phi_Y}{\partial q_0} & \frac{\partial \phi_Y}{q_1} & \dots & \frac{\partial \phi_Y}{q_n} & \frac{\partial \phi_Y}{d_1} & \dots & \frac{\partial \phi_Y}{d_n} \\ \frac{\partial \phi_Z}{\partial q_0} & \frac{\partial \phi_Z}{q_1} & \dots & \frac{\partial \phi_Z}{q_n} & \frac{\partial \phi_Z}{d_1} & \dots & \frac{\partial \phi_Z}{d_n} \end{pmatrix} \tag{19}$$

$$J(\vec{q}) = \begin{pmatrix} J_1(\vec{q}) \\ J_2(\vec{q}) \end{pmatrix} \tag{20}$$

En nuestro caso particular, obtenemos la siguiente matriz Jacobiana:

$$J_1(\vec{q}) = \begin{pmatrix} -\left(a_2 \cos(\theta_2) + a_3 \cos(\theta_2 - \theta_3) + d_1\right) \sin(\theta_1) & \left(-a_2 \sin(\theta_2) - a_3 \sin(\theta_2 - \theta_3)\right) \cos(\theta_1) & a_3 \sin(\theta_2 - \theta_3) \cos(\theta_1) \\ \left(a_2 \cos(\theta_2) + a_3 \cos(\theta_2 - \theta_3) + d_1\right) \cos(\theta_1) & \left(-a_2 \sin(\theta_2) - a_3 \sin(\theta_2 - \theta_3)\right) \sin(\theta_1) & a_3 \sin(\theta_1) \sin(\theta_2 - \theta_3) \\ 0 & a_2 \cos(\theta_2) + a_3 \cos(\theta_2 - \theta_3) & -a_3 \cos(\theta_2 - \theta_3) \end{pmatrix} \tag{21}$$

$$J_2(\vec{q}) = \begin{pmatrix} 0 & 1 & -1 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{pmatrix} \tag{22}$$

$$J(\vec{q}) = \begin{pmatrix} -\left(a_2 \cos(\theta_2) + a_3 \cos(\theta_2 - \theta_3) + d_1\right) \sin(\theta_1) & \left(-a_2 \sin(\theta_2) - a_3 \sin(\theta_2 - \theta_3)\right) \cos(\theta_1) & a_3 \sin(\theta_2 - \theta_3) \cos(\theta_1) \\ \left(a_2 \cos(\theta_2) + a_3 \cos(\theta_2 - \theta_3) + d_1\right) \cos(\theta_1) & \left(-a_2 \sin(\theta_2) - a_3 \sin(\theta_2 - \theta_3)\right) \sin(\theta_1) & a_3 \sin(\theta_1) \sin(\theta_2 - \theta_3) \\ 0 & a_2 \cos(\theta_2) + a_3 \cos(\theta_2 - \theta_3) & -a_3 \cos(\theta_2 - \theta_3) \\ 0 & 1 & -1 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{pmatrix} \tag{23}$$

Con la matriz Jacobiana ya obtenida se pueden obtener los puntos críticos. Dichos puntos permiten conocer posiciones inalcanzables por el brazo robótico, ya que el determinante en dicho punto es 0 y, por tanto, no existe la inversa.

Para este brazo robótico, el determinante es:

$$\det J(\vec{q}) = -a_2 a_3 \left(a_2 \cos(\theta_2) + a_3 \cos(\theta_2 - \theta_3) + d_1\right) \sin(\theta_3) \tag{24}$$

Analíticamente, los puntos críticos del μArm son: $\theta_3 = 0$ y $\theta_3 = \pi$. Pero, al observar las tablas con los ángulos de giro de los motores (ver tabla 1), comprobamos que el brazo robótico no es capaz de llegar a los 180° y, por la configuración geométrica del mismo, tampoco puede tener 0°[5], lo cual nos conduce a que no hay puntos críticos en este brazo robótico:

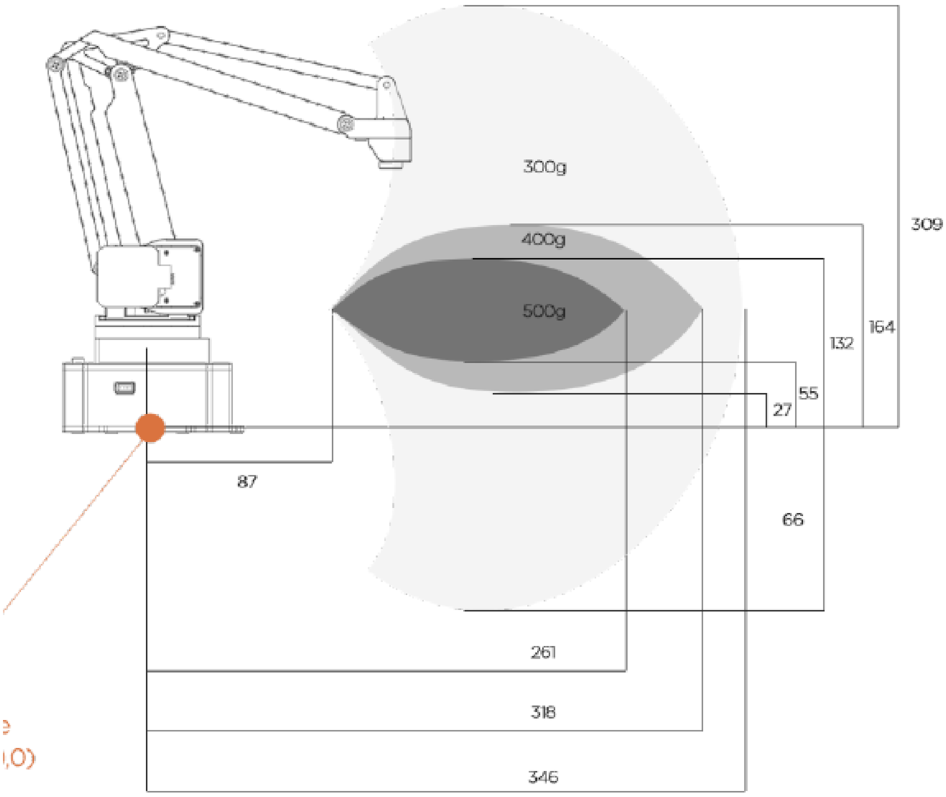

Figura 11: área de trabajo del μArm

5. Matriz Jacobiana inversa

$$\begin{pmatrix} -\frac{\sin(\theta_1)}{a_2\cos(\theta_2)+a_3\cos(\theta_2-\theta_3)+d_1} & \frac{\cos(\theta_1)}{a_2\cos(\theta_2)+a_3\cos(\theta_2-\theta_3)+d_1} & 0 \\ -\frac{\cos(\theta_1)\cos(\theta_2-\theta_3)}{a_2\sin(\theta_3)} & -\frac{\sin(\theta_1)\cos(\theta_2-\theta_3)}{a_2\sin(\theta_3)} & -\frac{\sin(\theta_2-\theta_3)}{a_2\sin(\theta_3)} \\ -\frac{(a_2\cos(\theta_2)+a_3\cos(\theta_2-\theta_3))\cos(\theta_1)}{a_2a_3\sin(\theta_3)} & -\frac{(a_2\cos(\theta_2)+a_3\cos(\theta_2-\theta_3))\sin(\theta_1)}{a_2a_3\sin(\theta_3)} & -\frac{a_2\sin(\theta_2)+a_3\sin(\theta_2-\theta_3)}{a_2a_3\sin(\theta_3)} \end{pmatrix}$$

[5]el motor tiene esos grados de libertad, pero la articulación no, y está limitada por la estructura

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).