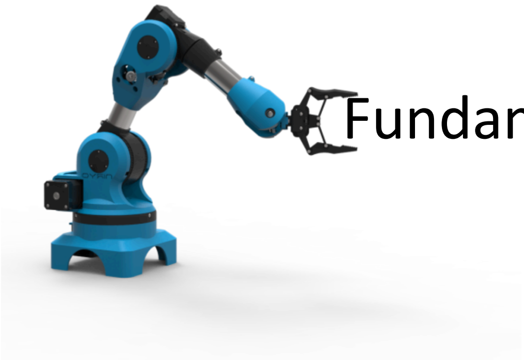

Fundamentos de Robótica Industrial

Cinemática Inversa

22 de Abril del 2025

Como en cada uno de los laboratorios, éste contará con una primera parte donde se realizará un control de lectura escrito, el cual tiene por objetivo, no solo verificar que conoce cómo se llevará a cabo el laboratorio, sino también afianzar conocimientos relacionados a la temática que trata esta instancia.

Laboratorio 2

Objetivo

Este laboratorio tiene por objetivo repasar los temas de localización espacial y cinemática de manipuladores vistos en el teórico, para afirmar los conocimientos a partir de su aplicación en cuestiones prácticas. En particular, se busca que los estudiantes apliquen la resolución de la cinemática inversa del manipulador antropomórfico de cuatro grados de libertad (ensamblado en el laboratorio 1) para mover el puntero del manipulador a un punto (x, y, z) deseado.

Fundamentos y Metodología

Antes de la realización del laboratorio es indispensable haber leído y tener presente la guía de “Buenas prácticas para trabajo de laboratorio”, disponible en el EVA del curso.

También es requisito que los estudiantes resuelvan la cinemática inversa del manipulador **previo** a asistir al laboratorio, para un mejor aprovechamiento del tiempo en el laboratorio.

Posición y orientación de un cuerpo rígido

Un cuerpo cualquiera se considera rígido si y sólo si para cualquier par de puntos que se tomen del mismo, la distancia entre ellos permanece inalterada ante la aplicación de cargas externas.

De esta forma, se pueden extender los conceptos de posición de una partícula, para hablar de “posición del cuerpo rígido”. Así, es habitual definir a posición de un cuerpo rígido a partir de las componentes de un vector que apunte a un punto en particular del cuerpo (p.ej.: el centro de masa), por lo tanto se tiene que son necesarios; 2 valores en un plano, o 3 valores en el espacio, para definir la posición. Por otro lado, la orientación de un cuerpo rígido admite varias representaciones, por ejemplo se pueden utilizar; las matrices de rotación, algún conjunto de ángulos de Euler, la representación de Eje-Ángulo, los cuaterniones, entre otros.

Cada una de estas representaciones tiene utiliza una cantidad de parámetros específica para definir la orientación de un cuerpo rígido, sujetos a un conjunto de restricciones. Dado que la representación mínima de la orientación de un cuerpo rígido en el espacio se obtiene a partir de 3 parámetros independientes, aquellas representaciones que utilicen más de 3 parámetros, deberán intrínsecamente presentar cierta interdependencia.

Específicamente, cada una de estas representaciones consta de las siguiente cantidad de parámetros y de restricciones:

- Matriz de rotación → 9 parámetros (matriz de 3x3)
  - 3 restricciones de ortogonalidad
  - 3 restricciones de normalidad (vectores unitarios)
- Ángulos de Euler → 3 parámetros
  - Representación mínima (existen varias combinaciones)
- Eje y ángulo → 4 parámetros (un eje de 3 coordenadas y un ángulo de giro)
  - 1 restricción de normalidad (vector unitario)
- Cuaterniones → 4 parámetros (variante del eje y ángulo)
  - 1 restricción de normalidad (vector unitario)

Manipulador Robótico

Un manipulador robótico se define como una cadena cinemática abierta que está compuesta por:

- Articulaciones cilíndricas y/o prismáticas
- Eslabones considerados como cuerpos rígidos
- Un extremo fijo y el otro libre

El movimiento de cada articulación puede ser de desplazamiento (prismática), de giro (cilíndrica), o una combinación de ambos. Cada uno de los movimientos independientes que puede realizar cada articulación con respecto a la anterior, se denomina grado de libertad (GDL).

En la práctica, en robótica sólo se emplean las articulaciones con un solo movimiento de rotación o de desplazamiento. En caso de que un robot tuviera alguna articulación con más de un grado de libertad, se podría asumir que se trata de varias articulaciones diferentes, unidas por eslabones de longitud nula.

Se define un cadena cinemática, como una serie de eslabones o barras unidas por articulaciones, por lo tanto, un manipulador como el de la Figura 1, que es una estructura mecánica, constituye una cadena cinemática, que además, es abierta porque uno de sus extremos se encuentra libre de restricciones.

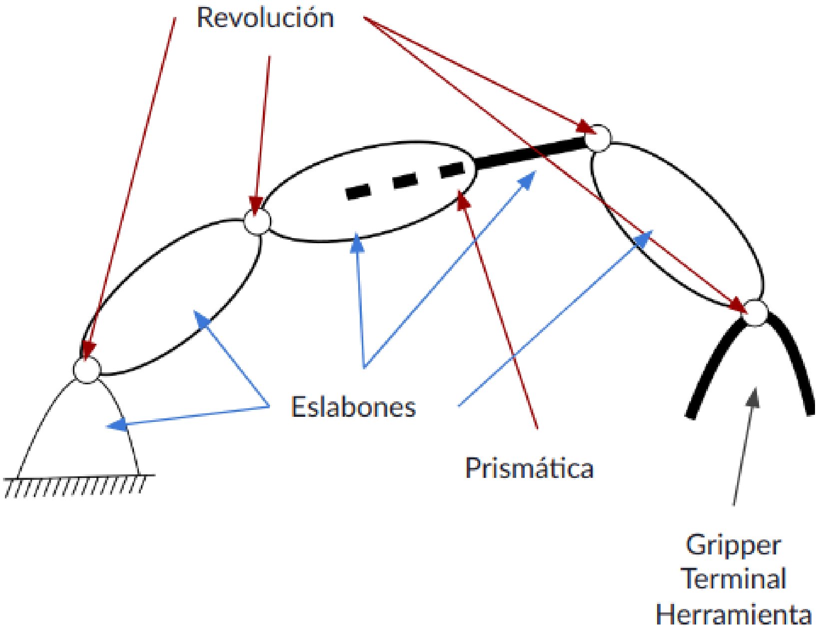

Figura 1: Esquema de manipulador genérico

Convención de Denavit-Hartenberg

Es un **método matricial** que establece la localización que debe tomar cada sistema de coordenadas *{S_i}* ligado a cada eslabón *i* de una cadena articulada, para poder **sistematizar** la obtención de las ecuaciones cinemáticas de la cadena completa.

Escogiendo los sistemas de coordenadas asociados a cada eslabón según la representación propuesta por D-H, será posible determinar la **Matriz de Transformación Homogénea (MTH)** $_{i\text{-}1}A_i$ que lleva un sistema de referencia al siguiente mediante **4 transformaciones básicas que dependen exclusivamente de las características geométricas del eslabón.**

$$_{i\text{-}1}A_i = \begin{bmatrix} C\theta_i & -C\alpha_i S\theta_i & S\alpha_i S\theta_i & a_i C\theta_i \\ S\theta_i & C\alpha_i C\theta_i & -S\alpha_i C\theta_i & a_i S\theta_i \\ 0 & S\alpha_i & C\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Si bien en general una matriz de transformación homogénea queda definida por 6 grados de libertad, recordando que son 3 GDL para definir la posición y 3 GDL para definir la orientación, el método de Denavit-Hartenberg, permite, en eslabones rígidos, reducir esta dependencia a 4 GDL con la **correcta elección de los sistemas de coordenadas**. Esto ocurre porque se considera que una articulación tan solo permite un GDL, por lo que la orientación del siguiente sistema de referencia, puede definirse como una rotación elemental del anterior (y por lo tanto, solo un parámetro asociado) si se seleccionan los sistemas de referencia convenientemente (convención de D-H).

Los pasos elementales para determinar los sistemas de referencias de un robot manipulador según la convención de Denavit-Hartenberg son los que se enlistan a continuación:

Numerar los eslabones en orden creciente, siendo **0** a la base fija del robot y **n** el último eslabón móvil

DH 1: Enumerar las articulaciones de forma creciente, siendo **1** el primer GDL y **n** el último.

DH 2: Localizar el eje de cada articulación. Si la art. es rotativa el eje será el eje de giro y si es prismática el eje será aquel en el que se produce el desplazamiento.

DH 3: Para **i** de **0** a **n-1**, situar el eje $\mathbf{z_i}$ sobre el eje de la articulación **i+1**

DH 4: Situar el origen del sistema de la base $\{S_0\}$ en "cualquier punto" del eje $\mathbf{z_0}$, mientras que los ejes $\mathbf{x_0}$ e $\mathbf{y_0}$ se ubicarán creando una terna ortonormal directa.

DH 5: Para **i** de **1** a **n-1,** situar el origen del sistema $\{S_i\}$ en la intersección del eje $\mathbf{z_i}$ con la línea normal común a $\mathbf{z_{i-1}}$ e $\mathbf{z_i}$. En particular, si los ejes se cortan, el origen se ubica en el punto de corte si los ejes son paralelos, el origen se sitúa en la articulación **i+1**

DH 6: Situar $\mathbf{x_i}$ en la línea normal común a $\mathbf{z_{i-1}}$ y $\mathbf{z_i}$

DH 7: Situar $\mathbf{y_i}$ de modo que forme un sistema ortonormal directo con $\mathbf{x_i}$ y $\mathbf{z_i}$

DH 8: Situar el sistema $\{S_n\}$ en el extremo del robot de modo que $\mathbf{z_n}$ coincida con la dirección de $\mathbf{z_{n-1}}$ y $\mathbf{x_n}$ sea normal a $\mathbf{z_{n-1}}$ y $\mathbf{z_n}$

Existen definiciones no únicas para los sistemas de referencia en los siguientes casos:

- Para el eslabón **0**, $\boldsymbol{O_0}$ y $\boldsymbol{x_0}$ se eligen arbitrariamente
- Para el eslabón $\boldsymbol{n}$**,** $\boldsymbol{z_n}$ no tiene una única definición mientras que $\boldsymbol{x_n}$ debe ser normal al eje $\boldsymbol{z_{n-1}}$.
- Cuando dos ejes consecutivos son paralelos, su normal común no es única.
- Cuando dos ejes consecutivos se intersectan, la dirección de $\boldsymbol{x_i}$ es arbitraria

Cinemática Directa

La resolución del problema cinemático directo implica la obtención de una expresión que permite conocer cuál es estado de la terminal, para cualquier valor que tomen las variables articulares.

Se utilizan principalmente dos estrategias para resolver un problema de cinemática directa:

- Métodos geométricos
- Matriz de transformación Homogénea

Métodos geométricos: La obtención de estas relaciones en ciertos casos (robots de pocos GDL) puede ser fácil de encontrar mediante simples consideraciones geométricas, teniendo en cuenta la disposición y características de las juntas y los eslabones.

Matriz de transformación Homogénea: Consiste en encontrar una MTH que relacione la localización espacial y orientación del extremo del robot con respecto al sistema de coordenadas de su base. Las entradas de esta matriz, serán funciones de las coordenadas articulares (***q***).

Cinemática Inversa

La resolución del problema cinemático inverso implica la obtención de una expresión para cada variable articular para que el estado de la terminal sea el deseado. La resolución de este problema es de especial importancia para transformar las especificaciones de movimiento desde el espacio operativo, hasta el espacio articular.

A diferencia de lo que pasa con el problema de cinemática directa, en el cual la terminal queda unívocamente determinada a partir de la posición de las articulaciones, no ocurre lo mismo con el problema cinemático inverso por las siguientes razones:

- Las ecuaciones a resolver son en general no lineales y por lo tanto no siempre es posible encontrar solución mediante una formulación cerrada.
- Pueden existir múltiples soluciones (codo hacia arriba - hacia abajo) o infinitas soluciones (redundancias)
- Pueden existir configuraciones no admisibles, lugares a los que la terminal no llega, restricciones de las articulaciones

Por lo tanto, la existencia de una solución está únicamente garantizada cuando el estado objetivo de la terminal pertenece al espacio de trabajo útil.

Se utilizan principalmente dos estrategias para resolver un problema de cinemática inversa.

- Métodos geométricos
- Matriz de transformación Homogénea

Métodos geométricos: Como es de esperar, este tipo de resolución implica la utilización de relaciones trigonométricas y geométricas sobre los elementos del robot. Se suele recurrir a la resolución de triángulos formados por articulaciones y eslabones. Esto implica que este tipo de métodos son adecuados para robots de pocos grados de libertad, o para aquellos que es posible considerar únicamente los primeros GDL.

Matriz de transformación Homogénea: A partir de la MTH de la cinemática directa, se podría encontrar las expresiones de las variables articulares en función de las variables espaciales. Es decir, conociendo las expresiones $f_{ij}(q)$ de la cinemática directa, y considerando las 6 restricciones de ortonormalidad de la matriz de rotación, se podría resolver el problema.

$$\mathbf{T} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & p_{x_0} \\ r_{21} & r_{22} & r_{23} & p_{y_0} \\ r_{31} & r_{32} & r_{33} & p_{z_0} \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} f_{11}(\mathbf{q}) & f_{12}(\mathbf{q}) & f_{13}(\mathbf{q}) & f_{x_0}(\mathbf{q}) \\ f_{21}(\mathbf{q}) & f_{22}(\mathbf{q}) & f_{23}(\mathbf{q}) & f_{y_0}(\mathbf{q}) \\ f_{31}(\mathbf{q}) & f_{32}(\mathbf{q}) & f_{33}(\mathbf{q}) & f_{z_0}(\mathbf{q}) \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\begin{array}{cc} r_{:,1}.r_{:,2} = 0 & r_{:,2}.r_{:,3} = 0 \\ r_{:,3}.r_{:,1} = 0 & ||r_{:,1}|| = 1 \\ ||r_{:,2}|| = 1 & ||r_{:,3}|| = 1 \end{array}$$

En la gran mayoría de los casos esto resulta impracticable en comparación con el método geométrico, o con métodos iterativos.

Desacoplamiento Cinemático

En manipuladores de 6 GDL, las primeras 3 variables articulares del robot son generalmente las que se utilizan para posicionar el extremo en las coordenadas (p_X, p_Y, p_Z) deseadas, y son habitualmente fáciles de obtener mediante el método geométrico.

Sin embargo, en la mayoría de los casos, no basta con posicionar el extremo del robot en un punto del espacio, sino que casi siempre es preciso también conseguir que la herramienta se oriente de una manera determinada. Para ello, estos robots cuentan con otros 3GDL, situados al final de la cadena cinemática y cuyos ejes, con frecuencia, se cortan en un punto, conocido como ***muñeca del robot***. Aunque una variación de estos tres últimos GDL origina un cambio en la posición final del extremo real del robot, su verdadero objetivo es poder orientar la herramienta del robot libremente en el espacio.

Teniendo en cuenta esto, se podría considerar que los 3 primeros GDL se utilizan para ubicar la muñeca del robot, y que los 3 restantes para orientar la herramienta (a menos de una distancia). A esto se le conoce como desacoplamiento cinemático y ofrece la ventaja de volver relativamente fácil de manejar el problema cinemático inverso.

Sin embargo hay que tener en cuenta que es **únicamente válido** cuando los ejes de las 3 últimas articulaciones se cortan en un punto (definido como muñeca).

Procedimiento de la práctica

Materiales a utilizar

Se utilizarán:

- Manipulador ensamblado en la práctica 1, cuya referencia se muestra en la figura 2
- USB 2D (figura 3)
- Fuente de alimentación 12V
- Computadora

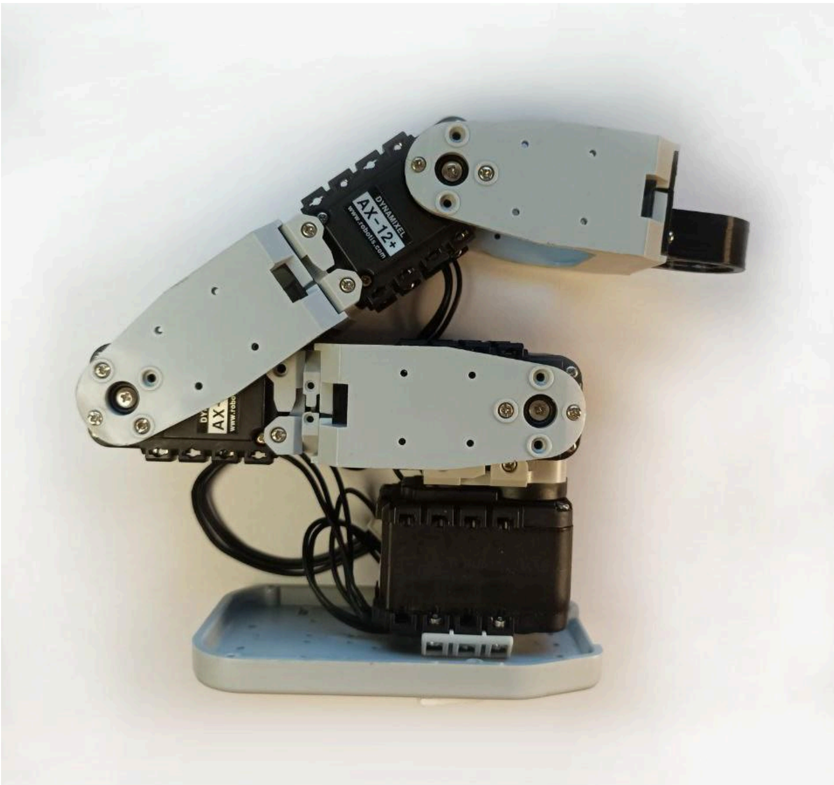

Figura 2: Referencia del manipulador a utilizar

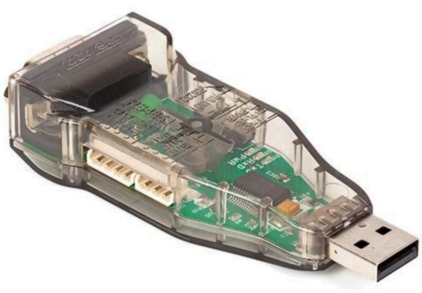

Figura 3: Módulo USB U2D2 de Robotis

Cinemática

Para la resolución de la cinemática inversa del manipulador se utilizará el método geométrico tomando como referencia el esquema de la figura 4, donde el sistema (x_0, y_0, z_0) estará fijo a la mesa de trabajo. Por otro lado, θ_1, θ_2, θ_3 y θ_4 serán los ángulos de giro de x_1 relativo a x_0, x_2 relativo a x_1, x_3 relativo a x_2 y x_4 relativo a x_3 respectivamente; mientras a_1, a_2, a_3 y a_4 son las distancias entre los orígenes de coordenadas como se ve en el esquema.

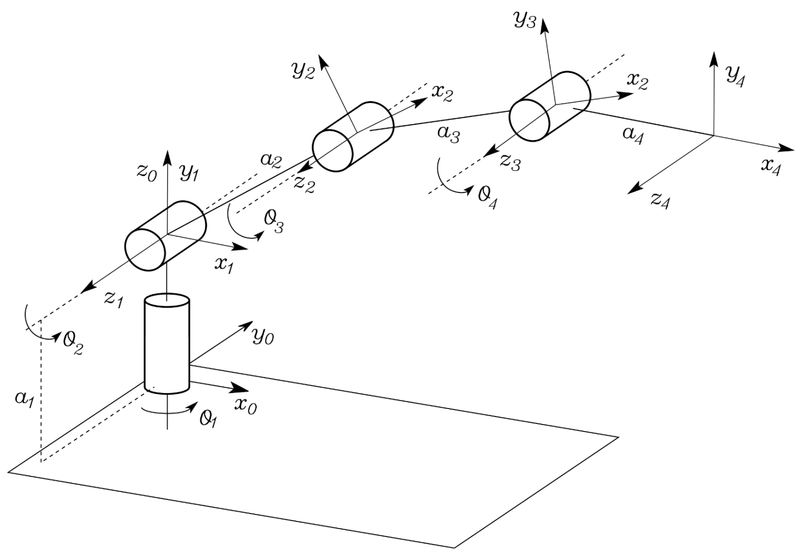

Figura 4: Esquema 3 GDL

El objetivo del problema de cinemática inversa es hallar las ecuaciones $\theta_1({}^0\mathbf{X}_4)$, $\theta_2({}^0\mathbf{X}_4)$, $\theta_3({}^0\mathbf{X}_4)$ y $\theta_4({}^0\mathbf{X}_4)$, siendo ${}^0\mathbf{X}_4=({}^0x_4,\ {}^0y_4,\ {}^0z_4)$ las coordenadas en el espacio del puntero (gripper) respecto al sistema de coordenadas de referencia (x_0, y_0, z_0).
Dichas ecuaciones permitirán obtener los valores requeridos $\theta_1({}^0\mathbf{X}_4)$, $\theta_2({}^0\mathbf{X}_4)$, $\theta_3({}^0\mathbf{X}_4)$ y $\theta_4({}^0\mathbf{X}_4)$ necesarios para mover el manipulador a la posición ${}^0\mathbf{X}_4$ deseada.

Para este caso se aplicará una restricción de modo que el puntero se mantenga horizontal al llegar a las coordenadas objetivo.

En la figura 5 (a y b) pueden verse las vistas vertical y horizontal del esquema del manipulador en la posición cero de referencia, en **c)** y **d)** de la figura 6 y 7 respectivamente, pueden verse en una posición arbitraria distinta a la de referencia.

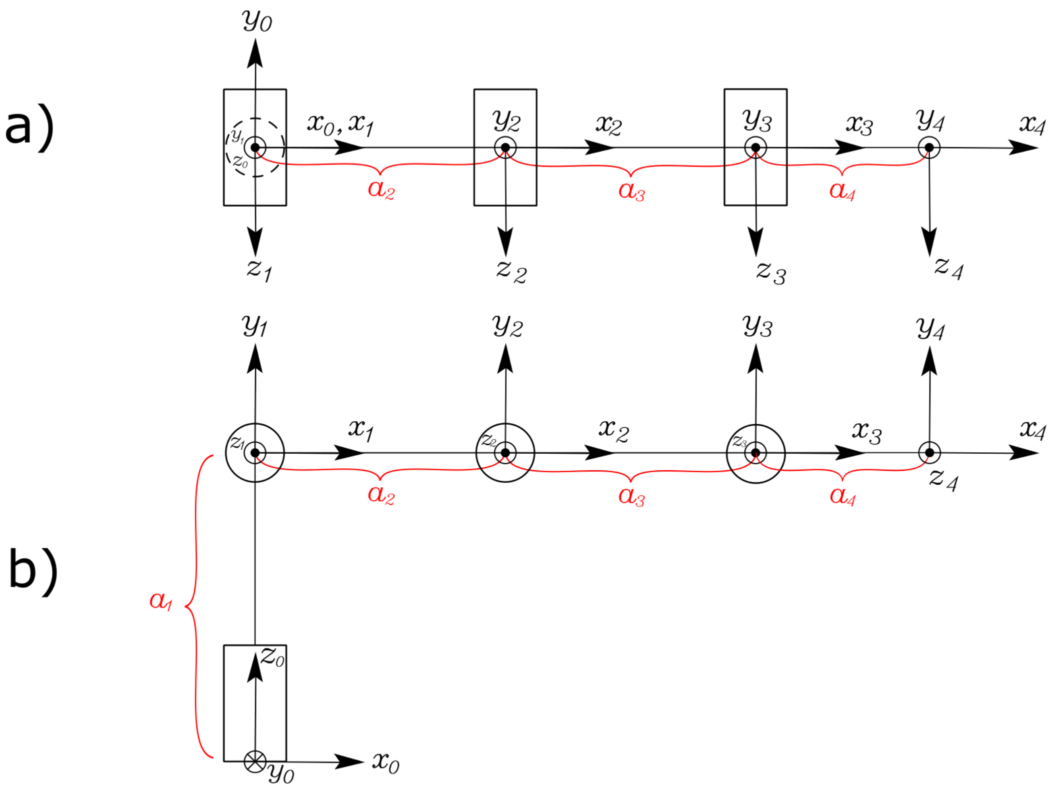

Figura 5: Esquema de vista vertical (a) y horizontal (b) del manipulador en posición cero

Para una mejor resolución de dicho problema se recomienda hacer uso de los parámetros auxiliares **Φ_1, Φ_2, Φ_3, r_1, r_2, r_3** y **r_4** presentes en en el esquema **d** de las figuras 6 y 7, así como del teorema del coseno.

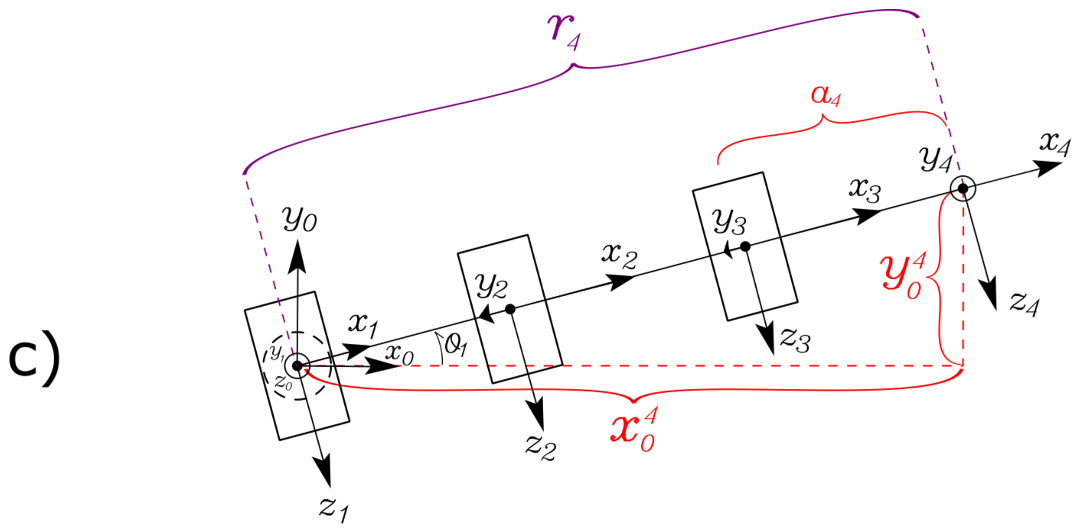

Figura 6: Esquema de vista vertical (c) y horizontal

d)

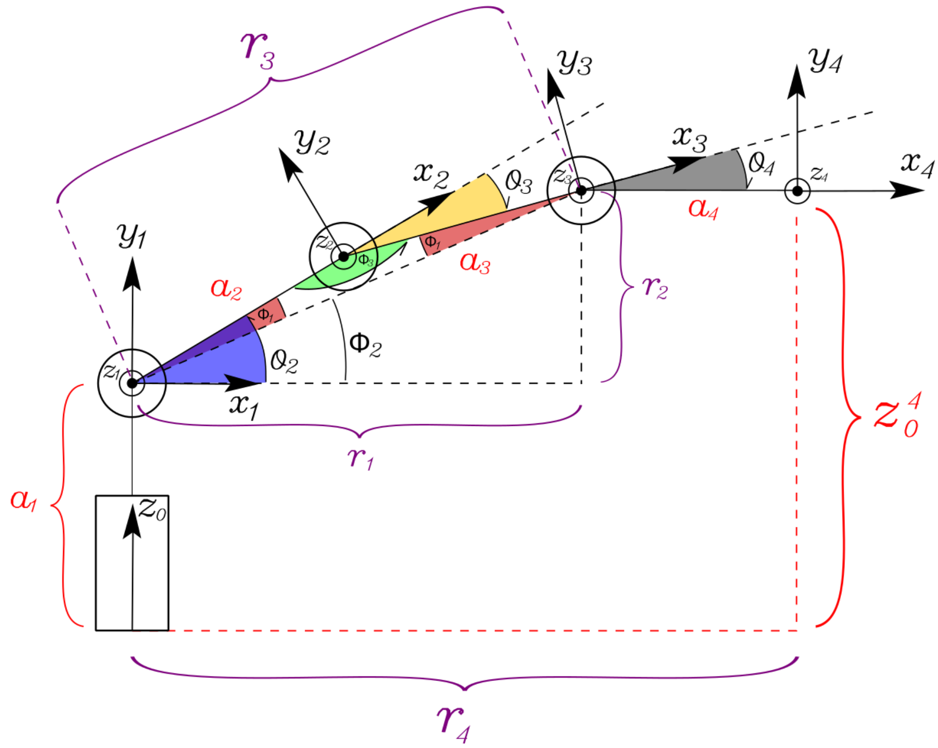

Figura 7: (d) del manipulador en posición arbitraria distinta de la posición cero

Código

Una vez obtenidas las relaciones entre la posición articular y la localización de la herramienta, se deberá:

1. Abrir Visual Studio (o editor de código preferido) y comenzar un archivo .py nuevo.
2. Aquí se deberá insertar el código correspondiente para θ_1, θ_2, θ_3 y θ_4 (ver anexo) de forma que, dado un punto en el espacio (x_4^0, y_4^0, z_4^0) en el cual se desea posicionar el puntero, el código resuelva cuánto deberá moverse cada servo.
3. En el mismo archivo, escribir el código correspondiente al movimiento del manipulador de forma que utilice los valores hallados θ_1, θ_2, θ_3 y θ_4
4. Guardar el archivo para luego correrlo en el Sandbox.

Utilizando el cuadriculado grabado en la estación de trabajo:

1. Posicionar cuidadosamente el gripper en su posición de referencia de forma que quede fiel al sistema de referencia, dibujar los ejes y ubicar el punto (x_4^0, y_4^0, z_4^0).
2. Medir cuidadosamente con la regla los valores de a_1, a_2, a_3 y a_4, teniendo en cuenta que el origen del sistema (x_4, y_4 z_4) se encuentra en el centro del orificio del puntero.

Habiendo colocado en el código los valores correspondientes y habiendo corroborado su funcionamiento, correr en el sandbox y verificar que el manipulador efectivamente alcance el punto (x_4^0, y_4^0, z_4^0) ± Error.

Evaluación del error

El error de un equipo al realizar una tarea o de un instrumento al efectuar una medida se define como la diferencia entre el valor obtenido $\hat{x}$ y el valor real x:

$$Error = \varepsilon = \hat{x} - x$$

En el caso de un manipulador se puede evaluar el Error como la diferencia entre el punto espacial que la terminal alcanzó y el valor objetivo.

El error en una medición se puede separar en dos componentes, el error sistemático y el error aleatorio.

Error sistemático (E_S): Este tipo de error se produce cuando hay un sesgo o una discrepancia constante entre el valor medido y el valor verdadero de la cantidad que se está midiendo. Este sesgo puede deberse a fallos en el equipo de medición, errores en el diseño del experimento o influencias externas que no se tienen en cuenta. Por ejemplo, si un manipulador siempre tiene una discrepancia en la misma dirección con respecto al punto deseado (por ej.: 1 cm a la derecha, o 3º en la orientación), esto sería un error sistemático.

Error aleatorio (E_A): Este tipo de error es inherentemente impredecible y fluctúa aleatoriamente con cada medición. Se debe a variaciones aleatorias en las condiciones de trabajo, la precisión limitada del equipo o el error humano al realizar el ensayo. A diferencia del error sistemático, el error aleatorio no sigue ningún patrón discernible y tiende a distribuirse alrededor de cero cuando se examinan muchos datos.

En resumen, el error sistemático afecta todas las mediciones de manera consistente y en una dirección particular, mientras que el error aleatorio introduce fluctuaciones aleatorias en las mediciones que pueden variar de una medición a otra. Ambos tipos de error son importantes de identificar y minimizar para obtener mediciones precisas y confiables.

Se define matemáticamente el error del posicionamiento del manipulador E_x cómo: $E_{\mathbf{x}} = E_S \pm E_A$

Además, considerando que se realizan *n* intentos de alcanzar el punto deseado, el error de cada medición *i,* es entonces:

$$E_{\mathbf{x}_i} = \mathbf{x} - \hat{\mathbf{x}}_i \qquad \forall i = 1, .., n$$

Finalmente, se estimarán las componentes del error como sigue:

El error sistemático con el sesgo: $E_S \leftarrow S = \mathbf{x} - \frac{1}{n}\sum_{i=1}^{n} \hat{\mathbf{x}} = \mathbf{x} - \bar{\mathbf{x}}$

Siendo el segundo término de la derecha de la igualdad ($\bar{x}$) la media del error.

El error aleatorio con la desviación estándar: $E_A \leftarrow \sigma = \sqrt{\frac{\sum_{i=1}^{n} ||\hat{\mathbf{x}}_1 - \bar{\mathbf{x}}||^2}{n-1}}$

Descripción de tareas:

1. Resolución de la cinemática inversa para el manipulador de 4 grados de libertad
2. Desarrollo de dicha resolución en Python
3. Conexión del manipulador a la PC y verificación de su funcionamiento
4. Elección de 1 punto en el espacio (x y z) y lograr que el manipulador alcance efectivamente dicho punto.
5. Evaluar el error del manipulador para n = 10 mediciones, completando la siguiente tabla.

Punto seleccionado: **X** = (,   ,   )

Medición	$\hat{x}_i=(p_x, p_y, p_z)_i$	$E_{xi}=\|\| \mathbf{X} - \hat{x}_i \|\|$	$\|\| \hat{x}_i - \bar{X} \|\|$
$\hat{x}_1$			
$\hat{x}_2$			
$\hat{x}_3$			
$\hat{x}_4$			
$\hat{x}_5$			
$\hat{x}_6$			
$\hat{x}_7$			
$\hat{x}_8$			
$\hat{x}_9$			
$\hat{x}_{10}$			
Media	$\rightarrow$	$\bar{X}$ =	$\sum_{i=1}^{n} \|\|\hat{\mathbf{x}}_1 - \bar{\mathbf{x}}\|\|^2 =$
Sesgo	$S = \mathbf{x} - \frac{1}{n}\sum_{i=1}^{n} \hat{\mathbf{x}} = \mathbf{x} - \bar{\mathbf{x}}$	S =	$\downarrow$
Desviación Estándar	$\sigma = \sqrt{\frac{\sum_{i=1}^{n} \|\|\hat{\mathbf{x}}_1 - \bar{\mathbf{x}}\|\|^2}{n-1}}$	$\rightarrow$	σ =

→Estimación de Error del manipulador:______________ ± ______________

Anexo

Instrucciones básicas de Python

Se utilizará la librería "math" de Python que por defecto trabaja con ángulos en radianes, tener en cuenta que los servomotores Dynamixel tienen un rango de 0º a 300º correspondientes a sus posiciones de 0 a 1023 mostradas en el Sandbox.

- `import math as m`

Permite trabajar con la librerìa "math" de Python utilizando el alias "m." para acceder a las distintas funciones implementadas en esta librería, por ejemplo: "m.atan(a/b)" devuelve el arcotangente de a sobre b. Para conocer todas las funciones disponibles en la librería "math" pueden referirse a: https://docs.python.org/2.7/library/math.html

Otros Ejemplos:

- `print(a)` → Imprime en terminal el valor de la variable a
- `print('a')` → Imprime en terminal el texto entre las comillas: a
- `print('texto descriptivo=',a)` → Muestra el valor de la variable a, seguido al texto descriptivo
- `chain.goto(id,posición,velocidad)` → Mueve el motor de id especificada a la posición y velocidad deseadas pero sólo puede moverse un motor a la vez

Opcional

En caso de querer lograr un movimiento más fluido puede llevarse a cabo moviendo todos los servos a la vez mediante las siguientes sentencias de código:

- ```
  initial_pos = {
  1: 530,
  2: 950,
  3: 60,
  4: 50,
  }
  ```

Define un diccionario con una posición determinada de cada uno de los servomotores, en el ejemplo mostrado se utilizó el nombre "`initial_pos`" pero puede definirse de cualquier otra forma así como pueden crearse varios diccionarios para varias posiciones.
  ```

- ```
  for i in range(4):
        chain.goto(i + 1, initial_pos[i + 1], speed, blocking=False)
  ```

La utilización de la estructura de control “`for`” en conjunto con la librería “HumanRobotics” para Dynamixel (en específico el método “`chain.goto`” pasándole el parámetro “`blocking=False`”) permite que todos los servos se muevan simultáneamente llevando el manipulador a la posición deseada ya que el parámetro “`blocking=False`” permite que el servo no deje de moverse para ejecutar la siguiente instrucción.

- `chain.wait_stopped()`

Si no se le pasan IDs por parámetro, espera a que todos los servos del manipulador se detengan. Es necesario después de alcanzar cada posición antes de moverse a la siguiente.

Ejemplo

```
speed=250
initial_pos = {
1: 545,
2: 990,
3: 190,
4: 350
}
final_pos = {
1: 600,
2: 900,
3: 150,
4: 400
}

for i in range(4):
  chain.goto(i + 1, initial_pos[i + 1], speed, blocking=False)

chain.wait_stopped()

for i in range(4):
  chain.goto(i + 1, final_pos[i + 1], speed=speed, blocking=False)

chain.wait_stopped()
```
```