



Tecnológico de Monterrey

Avance de proyecto

Sebastian Barrio Bejarano - A01753734

Abraham Trejo Arvizu - A01754195

Enrique Fest Flores - A01754194

Gabriela Marina Sánchez Ravelo - A01747124

Miércoles 28 de febrero del 2024

Diseño y desarrollo de robots MR 3001 B.102

Escuela de Ingeniería y Ciencias

Departamento de Mecatrónica

Investigación del estado de la técnica que soportó el desarrollo del presente sistema mecatrónico.	3
Descripción del sistema	9
Cinemática directa	11
Cinemática inversa	12
Administración del proyecto	14
Calendario de actividades	14
Conclusiones del proyecto	16
Referencias	18
Anexos	19

Investigación del estado de la técnica que soportó el desarrollo del presente sistema mecatrónico.

El sistema mecatrónico a desarrollar es un brazo robótico con una pinza, el cual debe priorizar la eficiencia y velocidad en el traslado de 10 dados, completamente iguales. El desarrollo de este sistema tendrá limitaciones con los materiales, los cuales fueron especificados por el socio formador. Sin embargo, los por menores del proyecto son a determinar por nosotros. Para ello daremos una revisión a la evolución en el área y observar que es lo que necesita nuestro proyecto.

El primer modelo de brazo robótico fue un acercamiento analógico dado por da Vinci, el cual estaba potenciado por la fuerza humana, después de ellos los avances no se materializarían hasta el siglo XX, donde el primer brazo robótico potenciado con motor nacería por un estudiante británico, Bill Taylor, y años después la industria tomaría manos en la obra y utilizaría los brazos para tareas repetitivas y pesadas, pero en los años consiguientes esta tecnología necesitaba de resultados precisos y programables para múltiples situaciones, así que se empezó a utilizar técnicas de cinemática en ellos, con el objetivo de que cada empresa pudiera obtener una de estas soluciones y aplicarlas a problemas específicos sin la necesidad de diseñar un producto único.

Un robot con tales funcionalidades, deberá de contar con diferentes aspectos Ingenieriles, estos serán, la aplicación de procesamiento de imágenes, para la segmentación de objetos cúbicos, Comunicación vía internet, para el control de un microprocesador, Modelación 3D sobre un brazo robótico, el estudio de su cinemática directa e inversa y la aplicación de la electrónica; Este documento se hablará de dos de ellas, Modelación 3D y cinemática directa e inversa.

Cinemática directa e inversa.

El estudio de la cinemática de manipuladores robóticos, está dividido en dos áreas, la cinemática directa y la cinemática inversa. La cinemática directa calculará la posición de la pinza basándose en los ángulos creados en las articulaciones del brazo, mientras que la cinemática inversa permitirá determinar los ángulos necesarios en las articulaciones para alcanzar posiciones específicas. Esto nos permitirá mover el brazo robótico a cualquier posición deseada usando un mapa de coordenadas, junto a la implementación de una cámara para determinar donde se encuentran los cubos en nuestro espacio.

La cinemática directa pide determinar un marco de referencia para cada una de las uniones de nuestro robot, las cuales pueden ser rotacionales o lineales, ya definido ellos, debemos de obtener las diferentes medidas de nuestros eslabones fijos y variables. Terminado ello, deberemos de aplicar nuestros resultados a una matriz de rotación y traslación, usando multiplicación de matrices, iremos desde cada eslabón a cada articulación multiplicando las matrices entre ellas en orden, y así obtendremos la matriz homogénea de nuestro robot.

Por otra parte, para resolver la cinemática directa se puede aplicar la convención de Denavit-Hartenberg, la cual utiliza matrices de transformación homogénea como una parte integral de su metodología. La principal diferencia entre este método y el de matrices de transformación homogénea es que en la convención D-H se asignan sistemas de referencia a cada uno de los eslabones y articulaciones, definiendo parámetros específicos para describir las relaciones espaciales entre ellos. Entre estos parámetros se incluyen longitudes, ángulos y desplazamientos, y son esenciales para caracterizar la configuración geométrica del robot.

A partir de estos parámetros se construyen matrices que describen la relación espacial entre cada uno de los sistemas de referencia, que se encuentran en los eslabones.

Existe una serie de pasos que se deben seguir para obtener los parámetros, los cuales son:

1. Numerar eslabones y articulaciones

- Eslabones se numeran desde 0 hasta n .
- Articulaciones se numeran desde 1 hasta n .

2. Localizar ejes de articulaciones

- Para revoluciones, el eje es el de giro.
- Para prismáticos, el eje es el de desplazamiento.

3. Colocar ejes Z

- Z_i se coloca sobre el eje de la i -ésima articulación.

4. Colocar sistema de coordenadas 0

- Se coloca el origen en algún punto a lo largo de Z_0 .

5. Colocar resto de sistemas de coordenadas

- Se colocan en las intersecciones de Z_i y la normal común a Z_i y Z_{i+1} .

6. Colocar ejes X

- Cada X_i va en la dirección de la normal común a Z_{i-1} y Z_i .

7. Colocar ejes Y

- Determinados por XYZ dextrógiro

8. Sistema del extremo del robot

- El sistema XYZ n se coloca en el extremo del robot.

9. Ángulos θ

- Cada θ_i es el ángulo de rotación alrededor de Z_i .

10. Distancias d

- Cada d_i es la distancia a lo largo de Z_{i-1} desde XYZ $_{i-1}$ hasta la intersección de las normales comunes.

11. Distancias a

- Cada a_i es la longitud de la normal común.

12. Ángulos α

- Ángulo de rotación alrededor de X_i para llegar de Z_{i-1} a Z_i .

13. Matrices de transformación de cada eslabón individuales

- Cada eslabón define una matriz de transformación utilizando los parámetros θ_i , d_i , a_i , y α_i .

14. Matriz de transformación final

- La matriz de transformación total que relaciona la base del robot con su herramienta es la encadenación (multiplicación) de todas esas matrices.

Para resolver el problema de cinemática inversa, se puede realizar a partir de diferentes métodos.

Método geométrico:

El primer método es el geométrico se basa en relaciones geométricas y trigonométricas entre las articulaciones y el extremo del robot. A partir de las ecuaciones obtenidas en la cinemática directa, se invierten para expresar las variables de articulación en función de la posición y orientación deseadas del extremo y se resuelven las ecuaciones, utilizando relaciones geométricas. Es importante mencionar que este método se limita a usarse en robots con pocos grados de libertad.

Método numérico:

Se utilizan métodos numéricos para que el error entre la posición del efector final de referencia y la posición actual sea lo mas mínimo posible, por lo que se consigue una aproximación. La inversa de la Jacobiana se puede utilizar para ésto, teniendo como desventaja que la inversa tiende a provocar saltos grandes haciendo el método numéricamente inestable, para que el método se pueda aplicar, la matriz jacobiana debe ser cuadrada. Para

resolver los problemas de la inversa de la Jacobiana, se puede hacer un cálculo utilizando la Pseudo-Inversa Moore - Penrose, éste cálculo es utilizado cuando la jacobiana no sea cuadrada y tenga más columnas que filas.

Un método alternativo sería el método de mínimos cuadrados amortiguados (o regularizados), en ésta, se utiliza un λ , que es un término de regularización, éste permite evitar la singularidad de la matriz jacobiana, cuando haya singularidad, la matriz será invertible y sea estable.

Método de matrices inversas:

Otro método para obtener la cinemática inversa, es el uso de matrices de transformación, el procedimiento nos pide que debamos de igualar nuestras matrices homogéneas a una matriz “normal, orientación, aproximación, posición”, dada la igualdad con álgebra eliminaremos una de las matrices con su inversa en ambos lados de la ecuación, dejando nuestra multiplicación de matrices con un inciso menos y en el lado de la matriz “n,o,a,p” junto a la inversa de la matriz eliminada, multiplicaremos estas matrices y las igualaremos a nuestras matrices homogéneas, esto con la idea de despejar nuestras incógnitas y variables para obtener como se definirán las rotaciones

Método por desacoplo cinemático:

Éste método es aplicable en robots con más de 3 grados de libertad, el objetivo va a ser encontrar el valor de las variables articulares que permiten que el robot alcance una posición y orientación específica en su efector final.

El método se basa en dividir en dos subproblemas, el posicionamiento y la orientación.

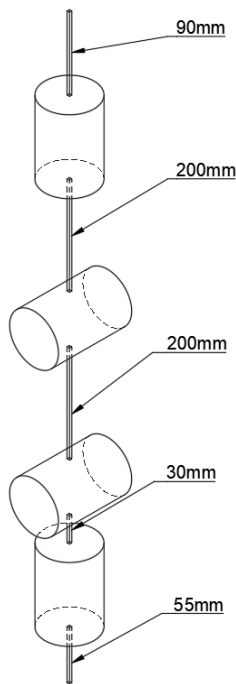
El subproblema de posicionamiento se resuelve utilizando técnicas geométricas o algebraicas para determinar las variables articulares que ubican la muñeca en la posición deseada, las

coordenadas articulares q_4 , q_5 , y q_6 , se pueden calcular únicamente con la información de la rotación deseada ($[noa]$),

La orientación (q_1 , q_2 y q_3) se obtiene de la información de la posición deseada (p_x , p_y y p_z), utilizando las mismas técnicas de posicionamiento, ésta etapa se realiza una vez se haya fijado la posición del efector final.

Descripción del sistema

En este reto se nos permitió utilizar un total de 5 servomotores, 3 de ellos siendo del modelo mg995 y los 2 restantes modelo sg90, utilizando un servo sg90 para abrir y cerrar el efector final, nos quedan 4 actuadores para el resto del robot, lo que significa que podemos tener hasta 4 grados de libertad. Para lograr un robot que pudiera posicionar el efector en cualquier lugar de su zona de trabajo y rotarlo en el eje de Roll del efector final se utilizó una



configuración hombro codo muñeca como la que se detalla a continuación.

El robot cuenta con 4 articulaciones revolutas:

La base (01) rota en el eje Z y está elevada 55mm, tiene una distancia desde el eje de rotación hasta el siguiente actuador de 30 mm.

El hombro (02) rota en el eje Y, se encuentra a 30 mm de la base

El codo (03) se encuentra a 200 mm del hombro, al final de lo que sería el primer eslabón, o el brazo, y también rota en el eje

Y, en la misma orientación que el hombro.

Por último, **la muñeca (04)** se encuentra a 200 mm del codo, estando al final del antebrazo, o eslabón 2 y este rota en el eje Z, permitiendo modificar el ángulo Roll del efector final.

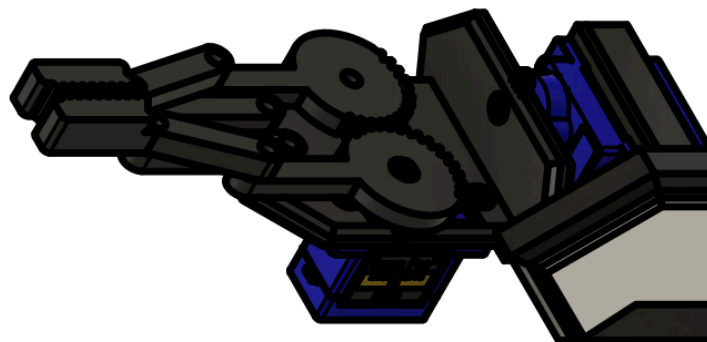


FIGURA 2. Efector final del robot

El efector final (Fig.2) es un gripper común que utiliza engranajes para transmitir el movimiento rotacional del servo a un movimiento de abrir y cerrar en las pinzas del gripper, el lugar donde las pinzas se encuentran cerradas se encuentra a 90 mm de la articulación de la muñeca.

Diseño (CAD)

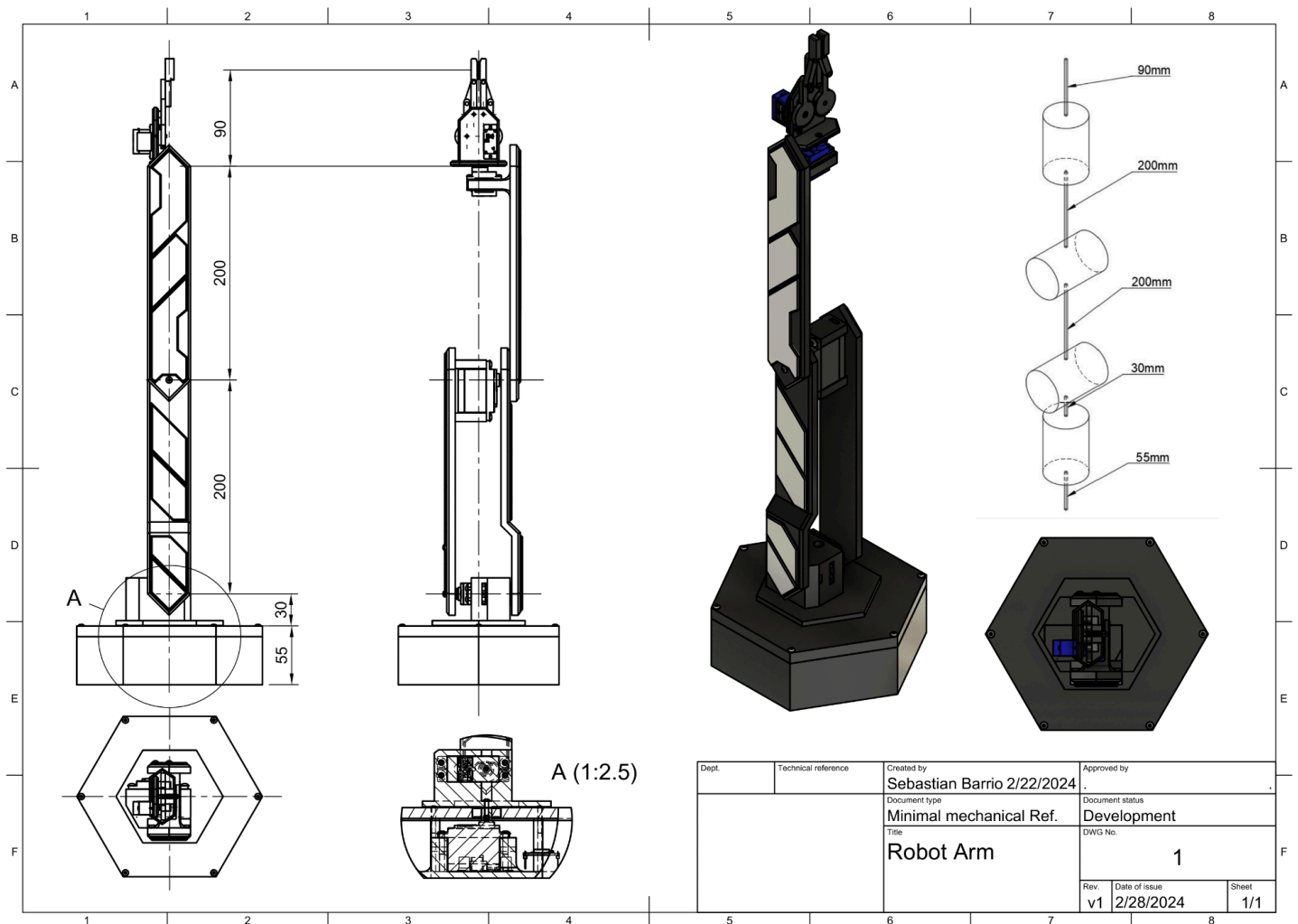


FIGURA 3. Dibujo técnico del cad, vista frontal, lateral, superior e isométrica, sección interna del servo de la base (A) y configuración cinemática ([link al cad](#))

Las especificaciones descritas se implementaron en el diseño del robot, contando con una base hexagonal donde se encuentran los microcontroladores y circuitos necesarios para la operación del robot y el servomotor de la base. Todas las uniones cuentan con un rodamiento

para disminuir fricción entre partes y evitar que los eslabones se trasladen de manera no deseada o se generen esfuerzos de más en los motores, se usan también tornillos M2, M3 y M4 para asegurar todas las partes que lo requieren. El material de construcción es PLA como material de impresión en 3D.

Cinemática directa

Para conseguir la cinemática directa del robot, se utilizaron las medidas y la configuración descritas anteriormente en la Figura 1, y se utilizó el método de las matrices de transformación homogéneas por su facilidad de uso y la simplicidad del modelo de robot que propusimos. Para nuestro robot definimos primero los 4 “eslabones” cada uno con su traslación y rotación correspondiente al actuador al que pertenecen, siendo entonces:

$${}^0T_1 = Trans(0, 0, a_1) * Rot(z, \Theta_1) * Trans(0, 0, a_2)$$

$${}^1T_2 = Rot(y, \Theta_2) * Trans(0, 0, a_3)$$

$${}^2T_3 = Rot(y, \Theta_3) * Trans(0, 0, a_4)$$

$${}^3T_4 = Rot(z, \Theta_4) * Trans(0, 0, a_5)$$

Donde el sistema de referencia 0 se encuentra en el punto más bajo del robot, teniendo en cuenta incluso la caja de electrónicos, y el sistema de referencia número 4 siendo el sistema de referencia final, que se encuentra apuntando en el mismo eje que el effector y que el eje Z universal, por lo que no hace falta ninguna rotación del sistema, esto nos da como matriz final de transformación:

$$T_F = {}^0T_4 = {}^0T_1 * {}^1T_2 * {}^2T_3 * {}^3T_4$$

Que al sustituir por las matrices correspondientes a las transformaciones, los valores de a1-a5 y hacer las multiplicaciones necesarias nos queda (el código utilizado se encuentra en el anexo):

$$T_F = \begin{bmatrix} -S(\theta_1)S(\theta_4) - C(\theta_4)(C(\theta_1)S(\theta_2)S(\theta_3) - C(\theta_1)C(\theta_2)C(\theta_3)) & S(\theta_4)(C(\theta_1)S(\theta_2)S(\theta_3) - C(\theta_1)C(\theta_2)C(\theta_3)) - C(\theta_4)S(\theta_1) & S(\theta_2 + \theta_3)C(\theta_1) & 10C(\theta_1)(29S(\theta_2 + \theta_3) + 20S(\theta_2)) \\ C(\theta_1)S(\theta_4) - C(\theta_4)(S(\theta_1)S(\theta_2)S(\theta_3) - C(\theta_2)C(\theta_3)S(\theta_1)) & C(\theta_1)C(\theta_4) + S(\theta_4)(S(\theta_1)S(\theta_2)S(\theta_3) - C(\theta_2)C(\theta_3)S(\theta_1)) & S(\theta_2 + \theta_3)S(\theta_1) & 10S(\theta_1)(29S(\theta_2 + \theta_3) + 20S(\theta_2)) \\ -S(\theta_2 + \theta_3)C(\theta_4) & S(\theta_2 + \theta_3)S(\theta_4) & C(\theta_2 + \theta_3) & 290C(\theta_2 + \theta_3) + 200C(\theta_2) + 85 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

FIGURA 4. Matriz de transformación final, obtenida con MATLAB.

Con esta matriz encontrada podemos encontrar el punto en el espacio en el que se encuentra el efector final si le damos los ángulos t1 a t4, que al comprobar con el modelo 3D y los ángulos igualados a 0 podemos observar que son correctos.

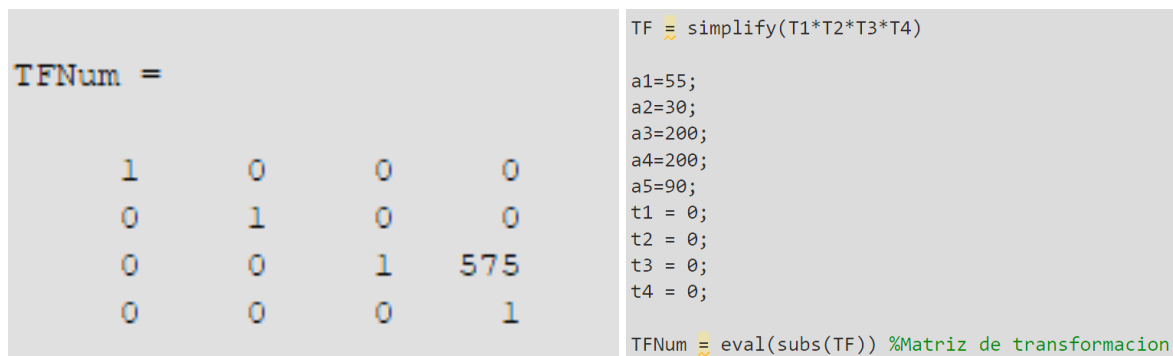


FIGURA 5. Comprobación de posición del efector final en MATLAB..

Cinemática inversa

Una vez que encontramos la cinemática directa ahora nos disponemos a encontrar la cinemática inversa del sistema, para esto decidimos utilizar el método geométrico por su simplicidad a la hora de calcularlo y el hecho de que llegamos a una solución analítica y simbólica que luego podemos implementar en el controlador una vez construido el robot, para esto primero definimos 2 planos, uno superior y uno lateral para poder resolver el problema por partes, para el plano superior xy proponemos lo siguiente para encontrar theta 1.

$$\frac{y}{x} = \tan \Theta_1$$

$$\Theta_1 = \tan^{-1}\left(\frac{y}{x}\right)$$

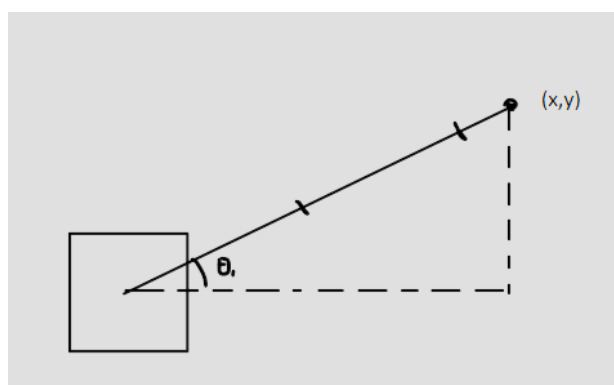


FIGURA 6. Plano superior XY, con la incógnita θ_1 .

Para encontrar theta 2 y theta 3 se utilizaron unos ángulos auxiliares phi1-phi3 y se utilizó la ley de cosenos para resolver para los angulos theta2 y theta 3 en configuración de codo superior, como se muestra en la figura siguiente

$$L = \sqrt{(x^2 + z^2)}$$

$$\phi_2 = \cos^{-1}\left(\frac{a_3^2 + (a_4 + a_5)^2 - L^2}{2(a_3)(L)}\right)$$

$$\phi_1 = \cos^{-1}\left(\frac{a_3^2 + L^2 - (a_4 + a_5)^2}{2(a_3)(L)}\right)$$

$$\phi_3 = \tan^{-1}(z/x)$$

$$\Theta_1 = \phi_3 + \phi_1$$

$$\Theta_1 = \phi_2 - \pi$$

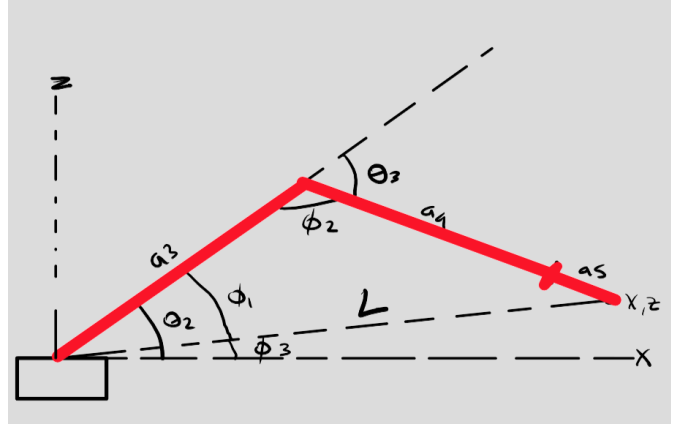


FIGURA 7. Plano ZX con 2 eslabones representados.

Y reemplazando obtenemos las siguientes ecuaciones, que junto con la ecuación encontrada en el plano superior nos permiten encontrar theta 2 y theta 3 en términos de las entradas x, y z

$$\Theta_1 = \tan^{-1}\left(\frac{y}{x}\right)$$

$$\Theta_2 = \tan^{-1}\left(\frac{z}{x}\right) + \cos^{-1}\left(\frac{a_3^2 + (x^2 + z^2) - (a_4 + a_5)^2}{2(a_3)(\sqrt{(x^2 + z^2)})}\right)$$

$$\Theta_3 = \cos^{-1}\left(\frac{a_3^2 + (a_4 + a_5)^2 - \sqrt{(x^2 + z^2)^2}}{2(a_3)(a_4 + a_5)}\right)$$

Una vez realizado esto podemos poner estas ecuaciones en matlab y de nuevo corroborar con el modelo 3D y podemos ver que son correctas.

<pre> %KINEMÁTICA INVERSA (por metodo geometrico)(resultado en radianes) syms px py pz px = 100; py = 100; pz = 100; %ecuaciones de cinematica inversa theta1 = atan(py/px); theta2 = atan(pz/px) + acos((a3^2+(px^2+pz^2)-(a4+a5)^2)/(2*a3*(sqrt(px^2+pz^2)))); theta3 = acos((a3^2+(a4+a5)^2-(sqrt(px^2+pz^2))^2)/(2*a3*(a4+a5))) - pi; %angulos conseguidos para un px py pz rad2deg(theta1) rad2deg(theta2) rad2deg(theta3) </pre>	<pre> ans = 45 ans = 160.2160 ans = -153.8202 </pre>
---	---

FIGURA 8. Código cinemática inversa en MATLAB.

Para este modelo de cinemática inversa no se calculo la theta 4, ya que como se puede apreciar en el diagrama y en la cinemática directa, la rotación de theta 4 no afecta la posición del efector final, solo la rotación del Roll del mismo.

Administración del proyecto

Tomando en cuenta que el proyecto debe desarrollarse en 5 semanas, se realizó un calendario con la distribución de actividades necesarias, con la finalidad de presentar un robot competente y eficiente para la competencia. Cada etapa del proyecto ha sido planificada, desde la definición inicial del proyecto hasta la participación en la competencia y la posterior documentación de resultados. La asignación de responsabilidades se definió de manera que haya una distribución equitativa del trabajo y una colaboración efectiva y productiva entre todos los integrantes del equipo.

Calendario de actividades

Semana	Actividad	Responsable
1	Revisión de reglamento de competencia y requisitos de construcción.	Abrham Trejo Sebastian Barrio
	Investigación de tecnologías para desarrollar el proyecto.	
	Investigación de los componentes y materiales.	

	Definición del prototipo objetivo.	Enrique Fest Gabriela Sánchez
	Planificación de actividades.	
	Establecimiento de roles y responsabilidades.	
2	Boceto de diseño mecánico	Sebastian Barrio
	Diseño en CAD	Sebastian Barrio y Gabriela Sánchez
	Cinemática directa	Enrique Fest
3	Cinemática inversa	Sebastian Barrio
	Impresión 3D de las piezas.	Sebastian Barrio y Enrique Fest
	Diseño del circuito electrónico.	Abraham Trejo
	Construcción del circuito electrónico.	Abraham Trejo y Enrique Fest
	Adaptación de contenedores y dados	Abraham Trejo y Enrique Fest
4	Ensamblaje de prototipo	Sebastian Barrio Gabriela Sánchez
	Integración de electrónica con el prototipo	Gabriela Sánchez
	Programación del controlador y actuadores	Sebastian Barrio
	Pruebas de interacción hardware-software	Abrham Trejo Sebastian Barrio Enrique Fest Gabriela Sánchez
	Programación de visión por computadora	Gabriela Sánchez Sebastian Barrio Abrham Trejo
	Prueba y Calibración del	

5	programa de visión por computadora	Abrham Trejo Sebastian Barrio Enrique Fest Gabriela Sánchez
	Pruebas y ajustes finales	
	Simulación del entorno de la competencia	
	Documentación del proyecto	

En este caso las actividades en las celdas verdes ya fueron realizadas y las blancas, son las próximas actividades que se llevarán a cabo durante la semana 3 a la 5. Lo que quiere decir que aún falta por realizar , la impresión en 3D del diseño, ensamble, circuito electrónico, programación de visión computacional y de actuadores y pruebas.

Conclusiones del proyecto

En la fase inicial del proyecto, se han logrado avances significativos, puesto que se desarrolló el diseño CAD del brazo robótico. La utilización de herramientas de modelado 3D ha sido fundamental para comprender la geometría y las interacciones entre los diferentes elementos del sistema, puesto que nos ha permitido, visualizar con precisión la estructura e incluso, la movilidad de los eslabones, de esta manera hemos podido analizar e identificar áreas de oportunidad, y así concluir si nuestro diseño es viable, y continuar con las siguientes fases.

Asimismo, al obtener la cinemática directa conseguimos una metodología para obtener la orientación y posición del efector final, que en este caso es la pinza, en función de las variables articulares, gracias a esto, tenemos una mejor comprensión de como se mueve el sistema. Por otro lado, al obtener la cinemática inversa podemos calcular las posiciones de las articulaciones necesarias para llegar a una orientación y posición específica, esto será de gran ayuda, pues el brazo podrá llegar a las diferentes posiciones para acomodar los dados correctamente. Ambas cinemáticas fueron comprobadas con el software de CAD donde sé

realizó nuestro diseño, y concluimos que con el diseño actual ambas cinemáticas fueron calculadas correctamente y son precisas.

Referencias

- A. Barrientos, L.F. Peñín, C. Balaguer, R. Aracil, «Fundamentos de robótica», McGraw Hill, 1997.
- Rodolfo Ponce R., Emmanuel A. Merchán C. , Luis Héctor Hernández G. , Celeste Salgado P., Arturo E. Flores Pa. (septiembre, 2022). “*Solución a la cinemática directa e inversa de manipuladores robóticos, empleando álgebra de cuaterniones duales*”.https://somim.org.mx/memorias/memorias2022/articulos/A3_33.pdf
- ESD/Robotics. “Historia y Evolución de la Robótica Industrial.” *EDS Robotics*, 2 February 2021, <https://www.edsrobotics.com/blog/evolucion-robotica-industrial/>. Accessed 28 February 2024.
- Moran, Michael E. “Evolution of robotic arms - PMC.” *NCBI*, 2007, <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4247431/>. Accessed 28 February 2024.
- Fikrul Akbar Alamsyah 2019 IOP Conf. Ser.: Mater. Sci. Eng. 494 012100

Anexos

CAD (disponible en variedad de formatos) : <https://a360.co/49vovwI>

Código de matlab:

```
clear all
clc
syms a1 a2 a3 a4 a5 t1 t2 t3 t4
syms nx ox ax px ny oy ay py nz oz az pz
T1 = Trans(0,0,a1)*Rot('z', t1)*Trans(0,0,a2);
T2 = Rot('y',t2)*Trans(0,0,a3);
T3 = Rot('y',t3)*Trans(0,0,a4);
T4 = Rot('z',t4)*Trans(0,0,a5);
TF = simplify(T1*T2*T3*T4)
a1=55;
a2=30;
a3=200;
a4=200;
a5=90;
t1 = 0;
t2 = 0;
t3 = 0;
t4 = 0;
TFNum = eval(subs(TF)) %Matriz de transformacion final dadas las medidas
%CINEMATICA INVERSA (por metodo geometrico)(resultado en radianes)
syms px py pz
px = 100;
py = 100;
pz = 100;
%ecuaciones de cinematica inversa
theta1 = atan(py/px);
theta2 = atan(pz/px) +
acos((a3^2+(px^2+pz^2)-(a4+a5)^2)/(2*a3*(sqrt(px^2+pz^2))));
theta3 = acos((a3^2+(a4+a5)^2-(sqrt(px^2+pz^2))^2)/(2*a3*(a4+a5))) - pi;
%angulos conseguidos para un px py pz
rad2deg(theta1)
rad2deg(theta2)
rad2deg(theta3)
%funciones auxiliares
function [RotMatrix] = Rot(axis, val)
    if axis == 'x'
        RotMatrix = [1,      0,      0,      0;
                     0,cos(val),-sin(val),0;
                     0,sin(val), cos(val),0;
                     0,      0,      0,      1];
    elseif axis == 'y'
        RotMatrix = [ cos(val),0,sin(val),0;
                     0,      1,      0,      0;
                    -sin(val),0,cos(val),0;
                     0,      0,      0,      1];
    elseif axis == 'z'
        RotMatrix = [ cos(val),-sin(val),0,0;
                     sin(val), cos(val),0,0;
```

```

                                0,      0,      1,0;
                                0,      0,      0,1];
    end
end
function [TrasMatrix] = Trans(x,y,z)
    TrasMatrix=[1,0,0,x;
                0,1,0,y;
                0,0,1,z;
                0,0,0,1];
end

```