

Proyecto final

Azamar Montero José Alberto, 201969464

Facultad de Ciencias de la Electrónica

Robótica I - Berenice Carreón Fernández

Benemérita Universidad Autónoma de Puebla

Abstract—La cinemática directa e inversa de un brazo robótico de 3 grados de libertad y la construcción y control del modelo que se manipula mediante 3 potenciómetros. Se establece una comunicación serial entre la placa Arduino UNO y Matlab para visualizar en tiempo real en pantalla las posiciones angulares (q_1, q_2, q_3) de cada motor y las coordenadas (x,y,z) del efecto final del robot.

Noviembre 21, 2023

I. INTRODUCCIÓN

El modelado y control de robots con eslabones rotacionales se simplifica gracias a la metodología/convención Denavit-Hartenberg con la que es posible relacionar los parámetros geométricos de un robot con su sistema de referencia base en cada eslabón que se sustituye en una estructura compacta llamada matriz homogénea en la que se presenta la rotación y el vector de traslación [1]. La cinemática inversa es el método ideal para ubicar el efecto final del sistema en coordenadas cartesianas específicas, con esto se define la cinemática inversa del robot antropomórfico de 3 grados de libertad (3gdl).

II. MARCO TEÓRICO

La cinemática directa es el estudio para determinar el movimiento de cada articulación respecto al eje de referencia que ubica el efecto final en el espacio de 3 dimensiones, el cual es un análisis de dicho movimiento en base a la geometría del robot. Este tipo de método requiere establecer un sistema de referencia fijo o base $\Sigma_0(x_0, y_0, z_0)$ relacionando la dependencia de las coordenadas articulares y los parámetros geométricos, que se expresa como:

$$\begin{bmatrix} x \\ y \\ z \\ \theta \\ \phi \\ \psi \end{bmatrix}^T = f_R(q)$$

Donde f_R es una función continua en el vector de posiciones o desplazamiento articular $q \in \mathbb{R}^n$ en la que n representa el número de grados de libertad y la dimensión del vector de posiciones o desplazamiento articular. $x_0, y_0, z_0 \in \mathbb{R}^3$ son las coordenadas cartesianas asociadas al extremo final del robot y θ, ϕ, ψ son los ángulos de Euler. Que representan la orientación de la herramienta colocada en el extremo final con respecto al sistema de referencia fijo en la base del robot.

El posicionamiento de un robot en el espacio tridimensional generalmente requiere de 6 coordenadas, 3 coordenadas representan la posición cartesiana del extremo final del robot

x_0, y_0, z_0 y 3 la orientación de la herramienta de trabajo denotadas por los ángulos de Euler θ, ϕ, ψ . Para el programador de robots en aplicaciones de automatización y procesos industriales, es mucho más fácil programar los movimientos del robot en coordenadas cartesianas que en coordenadas articulares. La cinemática directa realiza la función de convertir coordenadas articulares a coordenadas cartesianas.

Siempre es posible encontrar la cinemática directa, mientras que en la cinemática inversa pueden darse diversos escenarios, por ejemplo puede haber varias soluciones o no haber solución analítica, si este es el caso entonces pueden proponerse métodos numéricos, iterativos, geométricos, etc., como posibles formas de solución.

La cinemática inversa es un problema no lineal que relaciona las coordenadas articulares en función de las coordenadas cartesianas. Este problema representa la solución inversa de la cinemática directa: dada la posición cartesiana y la orientación de la herramienta colocada en el extremo final del robot, obtener los ángulos de las articulaciones. Aquí se busca definir las variables articulares (q_1, q_2, q_3) en función de la posición del efecto final, generalmente las ecuaciones para estimar los angulos se obtienen mediante el análisis gráfico del diagrama de cuerpo libre del robot.

$$q = f_R^{-1}(x, y, z, \theta, \phi, \psi)$$

Donde $f_R^{-1}(x, y, z, \theta, \phi, \psi)$ es función inversa de la cinemática directa [2].

La metodología Denavit-Hartenberg ofrece un procedimiento sencillo para obtener el modelo cinemático directo cuya estructura queda en representación de transformaciones homogéneas que se dan por el producto de cuatro transformaciones básicas:

$$H_i = R_{z,\theta_i}, T_{z,d_i}, T_{x,l_i}, R_{x,\alpha_i} = \begin{bmatrix} \cos(\theta_i) & -\sin(\theta_i) & 0 & 0 \\ \sin(\theta_i) & \cos(\theta_i) & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & l_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\alpha_i) & -\sin(\alpha_i) & 0 \\ 0 & \sin(\alpha_i) & \cos(\alpha_i) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} \cos(\theta_i) & -\sin(\theta_i)\cos(\alpha_i) & \sin(\theta_i)\sin(\alpha_i) & l_i\cos(\theta_i) \\ \sin(\theta_i) & \cos(\theta_i)\cos(\alpha_i) & -\cos(\theta_i)\sin(\alpha_i) & l_i\sin(\theta_i) \\ 0 & \sin(\alpha_i) & \cos(\alpha_i) & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

La cinemática diferencial directa es la derivada con respecto al tiempo de la cinemática directa

$$\frac{d}{dt}[x, y, z, \theta, \phi, \psi]^T = \begin{bmatrix} v \\ \omega \end{bmatrix} = \frac{d}{dt}f_r(q) = \frac{\partial f_R(q)}{\partial q}\dot{q} = J(q)\dot{q}$$

Donde se aprecia que la velocidad articular $\dot{q} \in \mathbb{R}^n$ con la velocidad lineal $v = \frac{d}{dt}[x, y, z]^T = [\dot{x}, \dot{y}, \dot{z}]^T \in \mathbb{R}^3$ y la velocidad angular $\omega = \frac{d}{dt}[\theta, \phi, \psi]^T = [\dot{\theta}, \dot{\phi}, \dot{\psi}]^T \in \mathbb{R}^3$, además el mapeo es descrito en términos de una matriz $J(q) = \frac{\partial f_R(q)}{\partial q} \in \mathbb{R}^{6 \times n}$ denominada jacobiano del robot o jacobiano analítico.

El jacobiano del robot representa una importante herramienta en robótica que sirve para caracterizar a un robot manipulador, encontrar configuraciones singulares, analizar redundancia, determinar la cinemática diferencial inversa, así como describir la relación entre fuerza aplicada y los pares o torques resultantes del extremo final [2]. Así pues, el determinante del jacobiano sirve para hallar el área de trabajo del robot mediante la delimitación de sus singularidades, es decir, los límites espaciales que puede alcanzar el efecto final.

III. DESARROLLO

Para el cálculo de la cinemática directa del robot antropomórfico de 3 grados de libertad (3gdl) primero se determina la configuración del robot mediante la tabla del método Denavit-Hartenberg, para este caso se define la tabla como sigue (las unidades son cm):

Eslabón	θ_i	β_i	l_i	α_i
1	q_1	14	1	$\pi/2$
2	q_2	2.15	19.2	π
3	q_3	1.55	12.7	0

Los valores se obtienen del análisis de los ejes β_i, l_i y los ángulos α_i , donde β es la dimensión del eje de los motores, medida sobre el eje z_{i-1} entre los ejes x_{i-1} y x_i , l_i la longitud del eslabón i -ésimo medida sobre los ejes x delimitado por los ejes z y α es el ángulo entre los ejes z_{i-1} y z_i medido con respecto al eje x_i . Estas definiciones se exponen de manera gráfica en la Figural.

De este modo se procede a estimar la matriz homogénea referente a esta configuración, para ello se utiliza un código en Matlab que se desarrolló en lo que va del curso:

```
clearvars; clc; %[ Rz Tz Tx Rx ]
t1=sym('q1'); b1=14; 11=1; a1=pi/2;
t2=sym('q2'); b2=2.15; 12=19.2; a2=pi;
t3=sym('q3'); b3=1.55; 13=12.7; a3=0;

H0_1 = H_DH(t1,b1,11,a1) ;
H1_2 = H_DH(t2,b2,12,a2) ;
H2_3 = H_DH(t3,b3,13,a3) ;
H0_3 = H0_1*H1_2*H2_3 ;

Ef = [H0_3(1,4);H0_3(2,4);H0_3(3,4)] ;
J = jacobian(Ef,[sym('q1') sym('q2') sym('q3')]) ;
J = simplify(J) ;
Det = simplify(det(J)) ;
```

Por cuestiones de practicidad se omite exponer los resultados de este código, donde $H0_3$ es la matriz homogénea del robot de 3gdl, Ef representa las coordenadas $[x; y; z]$ del efecto final, J el jacobiano que es una matriz de 3×3 y el determinante Det que al igualarlo a cero nos ayuda a determinar los límites de trabajo del robot. Al asignar valores

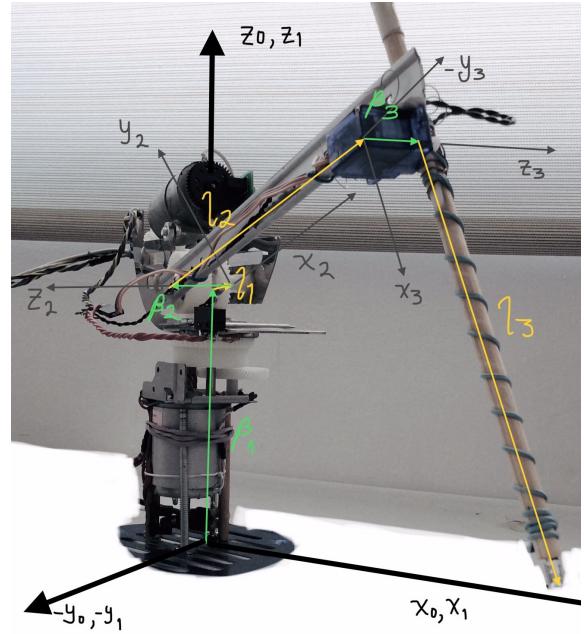


Fig. 1. Diagrama de los ejes del robot

para q_1, q_2, q_3 se determina una posición para el efecto final, el código continúa realizando una evaluación numérica de las variables simbólicas involucradas:

```
syms q1 q2 q3 b1 b2 b3 11 12 13 a1 a2 a3
simbolicos = [q1, q2, q3, b1, b2, b3, 11, 12, 13, a1, a2, a3];
numericos = [0,1.06,1.52,14 2.15,1.55,1,19.2,12.7,pi/2,pi,0];
H01 = subs(H0_1,simbolicos, numericos);
H02 = subs(H0_2,simbolicos, numericos);
H03 = subs(H0_3,simbolicos, numericos);

S1 = simplify([H01(1,4);H01(2,4);H01(3,4)]);
S2 = simplify([H02(1,4);H02(2,4);H02(3,4)]);
S3 = simplify([H03(1,4);H03(2,4);H03(3,4)]);
Ef = double(simplify([H03(1,4);H03(2,4);H03(3,4)]))

figure
plot3(0, 0, 0, 'o'); hold on
plot3([0,S1(1)], [0,S1(2)], [0,S1(3)],'r')
plot3(S1(1), S1(2), S1(3), x)
plot3([S1(1),S2(1)], [S1(2),S2(2)], [S1(3),S2(3)],'y')
plot3(S2(1), S2(2), S2(3), 'x')
plot3([S2(1),S3(1)], [S2(2),S3(2)], [S2(3),S3(3)],'b')
plot3(Ef(1), Ef(2), Ef(3), '*')

xlabel('x'); ylabel('y'); zlabel('z'); axis equal
legend("origen","Eslabon 1","Sigma 1","Eslabon 2",...
"Sigma 2","Eslabon 3","Efecto final")
grid on; hold off;
```

Así pues, el resultado de evaluar $[q_1, q_2, q_3] = [0, \pi, \pi/2]$ es $[x, y, z] = [-18.2, -0.6, 26.7]$ que se puede apreciar gráficamente en la figura 2. Nótese que el resultado en x es la suma escalar de $l_1 + l_2 = (1) + (-19.2)$. El resultado en y corresponde a la suma escalar de las dimensiones de los ejes de los rotores 2 y 3 denotados como $\beta_2 + \beta_3 = (2.15) + (-1.55)$.

El problema de la cinemática inversa se resuelve tomando como guía las ecuaciones para cinemática inversa de robots de 3gdl ampliamente conocidas en la literatura con β_2 y β_3 iguales a 0 y con longitud $l_1 = 0$. El problema para ajustar estas ecuaciones al modelo particular de robot con dimensiones $l_1 = 1, \beta_2 = 2.15, \beta_3 = 1.55$ radica en redefinir el sistema de referencia de los efectores 2 y 3. Las ecuaciones

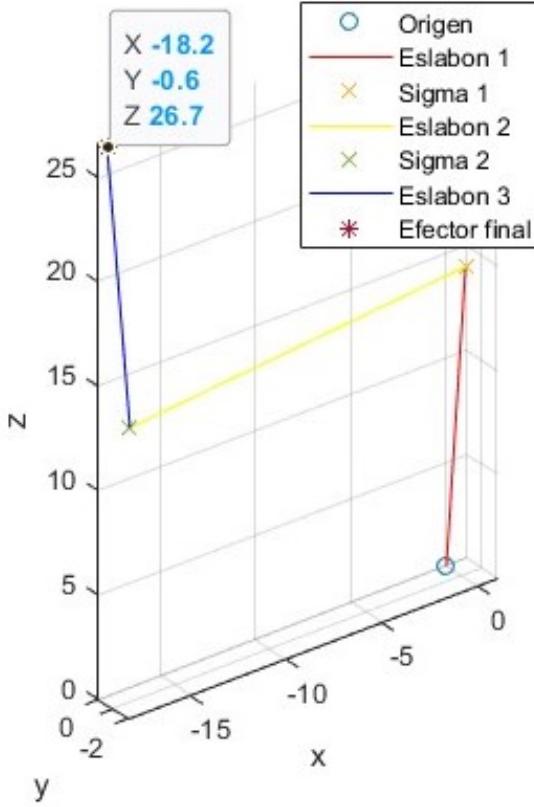


Fig. 2. Simulación de $[q_1, q_2, q_3] = [0, \pi, \pi/2]$

de cinemática inversa que se obtienen de [3] son las siguientes:

$$q_1 = \tan^{-1} \left(\frac{P_y}{P_x} \right) = \tan^{-1} \left(\frac{y}{x} \right) \quad (1)$$

Donde P_y y P_x corresponden a la posición del efecto final respecto de un sistema de referencia que coincide con el origen del segundo eslabón en el caso donde β_2, β_3 y l_1 son iguales a cero, del mismo modo P_z estaría desplazado respecto de estas ecuaciones cinemáticas. Enseguida se obtiene q_3 :

$$q_3 = \cos^{-1} \left(\frac{P_z^2 + P_x^2 + P_y^2 - (l_2^2 + l_3^2)}{2l_2l_3} \right) \quad (2)$$

Para el ángulo q_2 , se consideran dos casos donde el codo se encontrará sobre o debajo de la tangente que une el origen del hombro con el efecto final, para esto se definen las variables β y α :

$$q_2 = \beta + \alpha \rightarrow \text{codo arriba}$$

$$q_2 = \beta - \alpha \rightarrow \text{codo abajo}$$

En este caso se utiliza la configuración del codo arriba que, según mis apreciaciones, funciona bien dentro del espacio de trabajo con valores positivos de y , posteriormente se realizan correcciones para el resto del espacio de trabajo, así pues, q_2 se estima con:

$$q_2 = \tan^{-1} \left(\frac{P_z}{\sqrt{x^2 + y^2}} \right) + \tan^{-1} \left(\frac{l_3 \sin(q_3)}{l_2 + l_3 \cos(q_3)} \right) \quad (3)$$

A continuación se analizan los desplazamientos en x, y del hombro del robot realizando compensaciones a las ecuaciones

1,2 y 3. Las compensaciones son sobre la longitud del primer eslabón $l_1 = 1$ y $\beta_2 + \beta_3 = 0.6$, que son diferencias que desplazan el hombro (o el efecto final) 1 unidad en x y 0.6 en y para el caso donde los ángulos q_1, q_2, q_3 son cero. En la figura 3 se aprecia el caso de desplazamiento debido a β_2, β_3 de 0.6 unidades; para compensar el desplazamiento debido a l_1 se considera que el hombro se coloca sobre un círculo de 1cm de radio añadiendo así desplazamientos de $1\cos(q_1)$ para x y desplazamientos de $1\sin(q_1)$ para y , nótense que l_1, l_2, l_3 son paralelos en el plano x, y . La solución para q_1 con base en la figura 3:

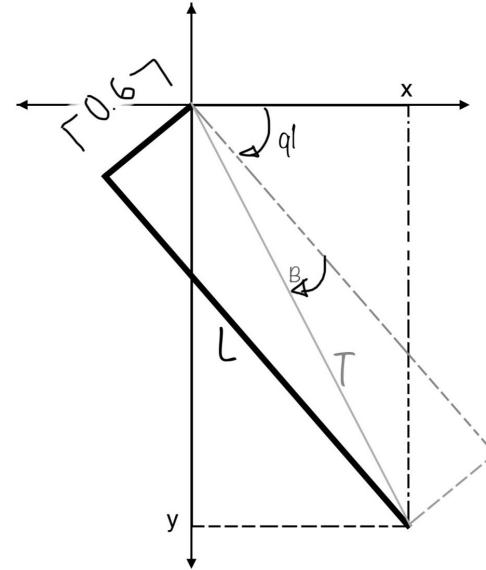


Fig. 3. Ajuste de desplazamiento de 0.6 centímetros del hombro

$$\begin{aligned} L^2 + 0.6^2 &= T^2 \wedge T^2 = x^2 + y^2 \\ L^2 + 0.6^2 &= x^2 + y^2 ; \quad L = T \cos(\beta) \\ \beta = \cos^{-1} \left(\frac{x}{T} \right) - q_1 &\rightarrow L = T \cos \left(\cos^{-1} \left(\frac{x}{T} \right) - q_1 \right) \\ T = \sqrt{x^2 + y^2}, L &= \sqrt{x^2 + y^2 - 0.6^2} \\ \cos \left(\cos^{-1} \left(\frac{x}{T} \right) - q_1 \right) &= \frac{L}{T} = \frac{\sqrt{x^2 + y^2 - 0.6^2}}{\sqrt{x^2 + y^2}} \\ q_1 = \cos^{-1} \left(\frac{x}{\sqrt{x^2 + y^2}} \right) - \cos^{-1} \left(\frac{\sqrt{x^2 + y^2 - 0.6^2}}{\sqrt{x^2 + y^2}} \right) \end{aligned}$$

Lo anterior sólo nos brinda el valor que debe tomar el ángulo q_1 , para hallar la compensación en x, y del desplazamiento de 0.6cm se evalúan por inspección jasja los límites de x, y encontrando que:

$$x \operatorname{sen}(q_1) - y \cos(q_1) = 0.6 \quad (4)$$

Esta ecuación 4 parece cumplirse para cualquier configuración de x, y, q_1 del robot de 3gdl lo cual es indicativo de que corresponde a la compensación total que considera a l_1 además de β_2, β_3 con lo que se tendría a la ecuación 1 siendo equivalente a la ecuación 4:

$$q_1 = \tan^{-1} \left(\frac{P_y}{P_x} \right) \rightarrow x \operatorname{sen}(q_1) - y \cos(q_1) = 0.6$$

Despejando los valores en x, y compensados denotados por P_x, P_y en la ecuación 1, a partir de la ecuación 4 al multiplicar por 1 el miembro derecho sabiendo que $1 = \sin^2(q_1) + \cos^2(q_1)$, se encuentra que:

$$\begin{aligned} q_1 &= \tan^{-1} \left(\frac{P_y}{P_x} \right) = \tan^{-1} \left(\frac{y - \sin(q_1) + 0.6\cos(q_1)}{x - \cos(q_1) - 0.6\sin(q_1)} \right) \\ P_y &= y - \sin(q_1) + 0.6\cos(q_1) \\ P_x &= x - \cos(q_1) - 0.6\sin(q_1); \end{aligned}$$

Aunado a la compensación $P_z = z - \beta_1$, estos ajustes de coordenadas corresponden a desplazar el sistema de referencia del sistema cinemático inverso obtenido de [3]. Ahora es posible evaluar las ecuaciones 1,2 y 3 con los valores P_x, P_y, P_z y obtener los ángulos que deben tomar los efectores para alcanzar las coordenadas x, y, z deseadas colocando el codo del robot arriba. Se escriben las ecuaciones en matlab:

```
x = 29; y = -0.6; z = 16;
if y < 0
    q1 = -acos(x/sqrt(x^2+y^2))...
+tacos(sqrt(x^2+y^2-(0.6*0.6))/sqrt(x^2+y^2));
else
    q1 = +acos(x/sqrt(x^2+y^2))...
+tacos(sqrt(x^2+y^2-(0.6*0.6))/sqrt(x^2+y^2));
end
x = x-cos(q1)-0.6*sin(q1);
y = y-sin(q1)+0.6*cos(q1);
z = z-14;
l2 = 19.2; l3 = 12.7;
q3 = acos((z^2)+x^2+y^2-(l2^2)+(l3^2))/(2*l2*l3);

xy = sqrt((x^2)+(y^2));
B = atan(z/xy);
a = atan((l3*sin(q3))/(l2+(l3*cos(q3))));
q2 = B+a;
```

Fue necesario incluir el ciclo if, ya que al evaluar la posición en ciertos octantes, el efecto resultante aparecía rotado 180°. Así pues, al evaluar las posiciones $[x, y, z] = [29, -0.6, 16]$ se obtiene el resultado $[q_1, q_2, q_3] = [0, 26.66, 58.05]$, si se evalúan estos valores de q_1, q_2, q_3 en el modelo cinemático directo, se obtienen nuevamente las coordenadas propuestas $[29, -0.6, 16]$ [cm] comprobando así, que el modelo cinemático inverso hallado es correcto.

IV. IMPLEMENTACIÓN CON EVIDENCIA

El primer enfoque para el diseño del robot, se vió influenciado por el hardware de control con el que ya se contaba, que se conforma por un puente H modelo L298N y una placa Arduino UNO la cual cuenta con dos entradas para interrupciones por hardware, lo que a su vez limita el número de motores con encoders incrementales que se pueden controlar, siendo 2 el máximo; con esto se decidió colocar un servomotor como tercer eslabón a controlar, ya que los servomotores cuentan con su propia estructura de control y sólo se requiere una señal PWM para controlarlos. Se acondicionaron con encoders 2 motores de DC obtenidos de impresoras viejas y se les colocaron mecanismos de motorreducción para aumentar la resolución de los motores. Los cables son reutilizados de sistemas viejos, al igual que los ejes, engranajes, armazón y eslabones, el servomotor es nuevo.

Para cumplir los objetivos de mostrar en tiempo real los ángulos de los motores junto a la posición del efecto final, se programó el microcontrolador para que el sistema transmita

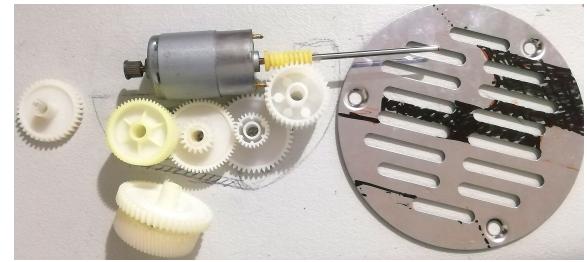


Fig. 4. Posible disposición de los engranajes

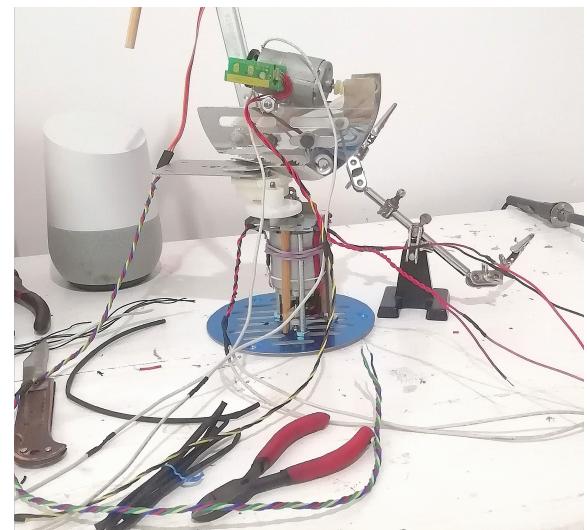


Fig. 5. Cableando el robot

vía serial los valores de los potenciómetros que controlan los motores, mientras que en Matlab se programó una estructura para leer el puerto serial y actualizar los últimos valores registrados en un gráfico además de evaluarlos en la estructura de cinemática directa e igualmente actualizando el gráfico con los últimos valores evaluados. El gráfico que se obtiene se muestra en la figura 6.

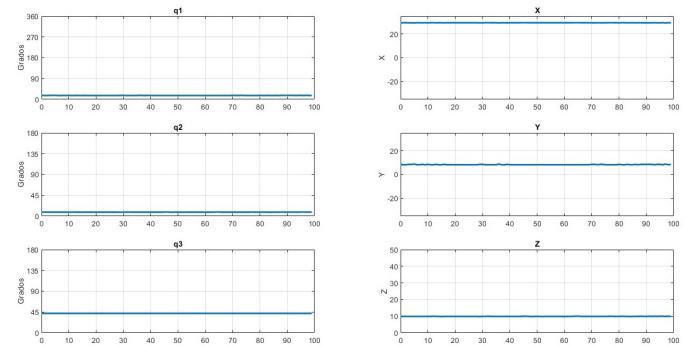


Fig. 6. Impresión de los últimos 100 datos recibidos y procesados vía serial

Adicionalmente se programó al robot para que ubicase su efecto final en los vértices de un cuadrado de 9cm de lado, este enciende un led en el efecto final cada que llega a la posición deseada, espera un segundo y continúa al siguiente vértice en un bucle infinito. Las posiciones angulares que se evalúan son:

```

double A[] = {0, 44.7, 110};
double B[] = {0, 26.66, 58};
double C[] = {-17.2, 21.43, 44};
double D[] = {-24.09, 41.52, 99};

```

Se puede apreciar el funcionamiento del robot y la impresión de datos en pantalla en el video de YouTube de la referencia [4].

V. CONCLUSIÓN

El método matemático aplicado para obtener la posición del efecto final se basa en las rotaciones y traslaciones/deplazamientos de sistemas de referencia, es genial apreciar cómo pueden controlarse los sistemas a través de sus modelos matemáticos, por ejemplo si se desea desplazar el efecto final sobre un círculo o alguna otra forma irregular sería necesario evaluar la cinemática inversa para cada coordenada de la forma o figura deseada, sin los modelos matemáticos esto sería muy difícil. El control del robot en este caso resultó simple, pues la robótica tiene un amplio repertorio de métodos para hacer ultra precisos y coordinados los movimientos de los robots, como lo pueden demostrar los proyectos de ciertas empresas de tecnología. No estoy seguro qué tan sofisticados son en la actualidad los robots con el certificado hecho en México, pero estoy seguro que sin el dominio de estos modelos sería imposible desarrollarlos.

REFERENCES

- [1] Siciliano, B., & Khatib, O. (Eds.). (2016). Springer handbook of robotics. Springer Science & Business Media.
- [2] Reyes Cortes, F., Cortes, F. R., Acuna, A. L., Acuna, A. L., Lopez Orozco, J. A., Vargas Ruiz, P.G., ... & Sugabaker, D. J. B. (2012). MATLAB aplicado a Robotica y Mecatronica (No. 681.51 670.4272). e-libro, Corp..
- [3] Barrientos, A., Peñín, L. F., Balaguer, C., & Aracil, R. (2007). Fundamentos de robótica. MC GRAW HILL.
- [4] <https://youtu.be/2ueuRCYDbU4>