



Universidad Veracruzana

Facultad de Ingeniería de la construcción y el habitat

TESIS:

**Diseño y construcción de un brazo robótico
colaborativo para sistemas de manufactura flexible**

Presenta Ángel Ernesto Trujillo Elizondo, para obtener el grado de
Maestría en Ingeniería Aplicada

Asesor:

José Alejandro Vasquez Santacruz

Índice general

Agradecimientos	5
Resumen	6
Abstract	7
1. Introducción	8
1.1. Objetivos	8
1.2. Justificación	9
1.3. Hipótesis	9
2. Marco teórico	10
2.1. Brazos robóticos	10
2.2. Robótica colaborativa	10
2.3. Sistemas de manufactura flexible	10
2.4. Ingeniería de sistemas basado en modelos	10
3. MBSE	11
3.1. Diagrama de requerimientos	11
4. Modelo matemático	13
4.1. Cinemática directa e inversa	15

4.1.1. Cinemática directa	15
4.1.1.1. Matriz de transformación homogénea	15
4.1.1.2. Convención de Denavit-Hartenberg	17
4.1.2. Cinemática inversa	17
4.2. Cinemática de la velocidad	18
4.3. Modelo dinámico	18
4.3.1. Formulación Newton-Euler	18
5. Componentes eléctricos	20
6. Componentes mecánicos	21
6.1. Actuador	21
6.1.1. Engranajes planetarios	22
6.1.2. Engrane cicloidal	23
6.1.3. Harmonic Drive	23
Bibliografía	24
Anexos	25
Anexo 1. Función de creación del brazo robótico	25
Anexo 2. Cinemática directa	27
Anexo 2. Cinemática directa	27
Anexo 2. Dinámica	28

Índice de figuras

3.1. Diagrama de requerimientos	12
4.1. Boceto del brazo robótico propuesto	14
4.2. Cadena cinemática	16
6.1. OpenTorque Actuator por Gabrael Levine	22

Índice de cuadros

4.1. Parámetros del brazo robótico	14
4.2. Parámetros Denavit Hartenberg	17
4.3. Parámetros del brazo robótico	19

Agradecimientos

Agradezco al Consejo Nacional de Ciencia y Tecnología por el apoyo económico recibido durante la realización de este posgrado.

Agradezco profundamente a mi asesor, el Dr. José Alejandro Vásquez Santacruz por su apoyo y paciencia durante la realización de esta investigación.

Resumen

Aquí va un resumen, pero cuando acabe.

Abstract

Here goes an abstract, but i'll do it when I finish.

Capítulo 1

Introducción

Los robots colaborativos, también denominados *cobots*, son aquellos que permiten a los humanos ocupar la misma área de trabajo que éstos y ofrecer la interacción segura entre robot y humano con el fin de realizar una tarea común.

Los *cobots* ofrecen mucho más flexibilidad en su operación con respecto a los robots industriales tradicionales, por lo que es posible su utilización en la manufactura de productos personalizados en masa. Las desventajas consisten en sacrificar la carga útil máxima así como la eficiencia y exactitud de los robots industriales tradicionales. En resumen, los robots colaborativos se encuentran en el punto medio entre el trabajo manual y la automatización industrial. [1]

El interés en los robots colaborativos ha ido en aumento en la última década, es por esto que los grandes fabricantes de robots como ABB, KUKA o Universal Robots han lanzado productos que satisfacen esta necesidad. Por su parte, las grandes empresas de manufactura como Audi, Volkswagen y Nissan han incorporado robots colaborativos en sus líneas de ensamblaje [1]

1.1. Objetivos

El objetivo general de esta investigación consiste en diseñar y construir un brazo robótico colaborativo de seis grados de libertad para sistemas de manufactura flexible.

Los objetivos específicos se mencionan a continuación:

- Desarrollar el modelo matemático cinemático y dinámico de un brazo robótico de seis grados de libertad.
- Construir un brazo robótico de coste accesible y fácil fabricación, así como compartir su diseño y componentes con una licencia de código abierto.
- Optimizar el modelado de piezas para su correcta manufactura con máquinas de manufactura aditiva.
- Desarrollar ¿utilizar? una metodología apegada a la ingeniería de sistemas basadas en modelos.

1.2. Justificación

El desarrollo de un brazo robótico de seis grados de libertad para sistemas de manufactura flexible que se pretende realizar es relevante en diversos ámbitos del conocimiento.

Por último, al estar pensado como un desarrollo de código abierto, contribuirá al acervo tecnológico de la humanidad y podrá ser copiado, modificado y mejorado alrededor del mundo.

1.3. Hipótesis

Capítulo 2

Marco teórico

2.1. Brazos robóticos

De acuerdo con Spong et al. [2, p. 1], la mayoría de las aplicaciones en el campo de la robótica se centran en brazos robóticos industriales que operan en fabricas con entornos estructurados, es por esto su gran importancia.

Un brazo robótico está compuesto de eslabones conectados por articulaciones para formar una cadena cinemática.

2.2. Robótica colaborativa

2.3. Sistemas de manufactura flexible

2.4. Ingeniería de sistemas basado en modelos

Capítulo 3

MBSE

3.1. Diagrama de requerimientos

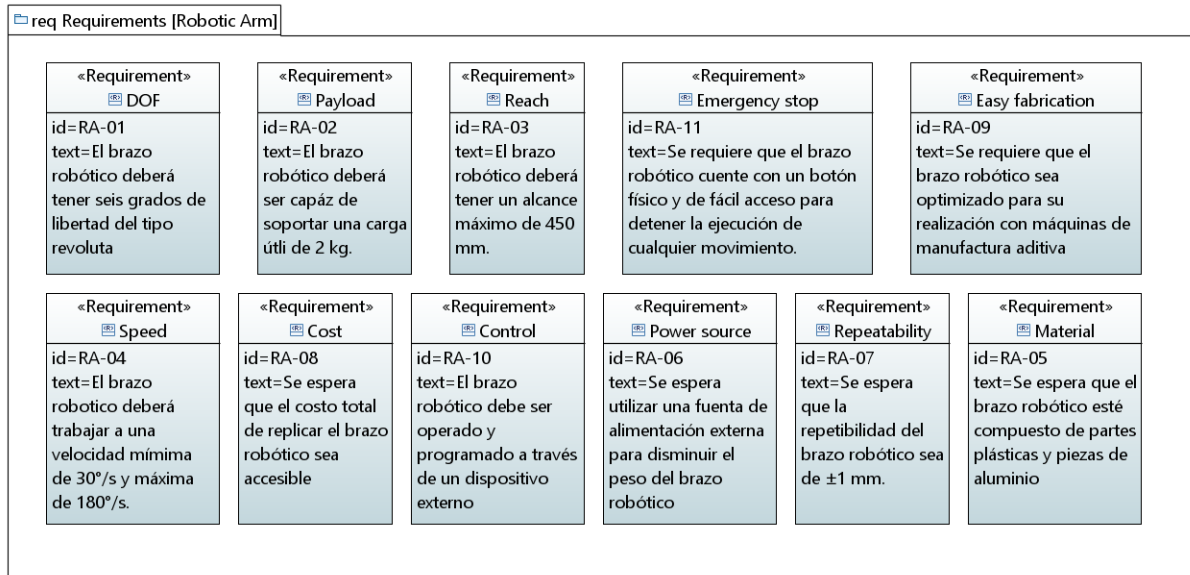


Figura 3.1: Diagrama de requerimientos

Capítulo 4

Modelo matemático

De acuerdo con [3, p. 14], el diseño y control de robots requiere diversos modelos matemáticos, tales como:

- Cinemática directa e inversa, es decir, encontrar la posición del efector final en términos de las coordenadas de las articulaciones y viceversa.
- Cinemática de la velocidad, encontrar la velocidad del efector final en términos de la velocidad de las articulaciones y viceversa.
- Modelo dinámico, el cual establece la relación entre los torques o fuerzas que ejercen los actuadores y las posiciones, velocidades y aceleraciones de las articulaciones.

En este capítulo se desarrollarán los modelos matemáticos necesarios para simular y diseñar el brazo robótico, así como predecir el comportamiento del mismo.

Para realizar estos modelos, es importante contar con los parámetros físicos y geométricos del robot, los cuales aún no están completamente definidos, por lo que, en una primera aproximación, se mencionan en la tabla 4.1.

Otros requerimientos necesarios para el desarrollo del modelo matemático es el alcance total del brazo, el cual deberá ser de mínimo 500 mm, la velocidad, la cuál deberá estar en un rango entre 5 RPM y 30 RPM, y por último, la carga útil deberá ser de 2 kg.

En la figura 4.1 podemos ver un boceto del brazo robótico que se planea implementar.

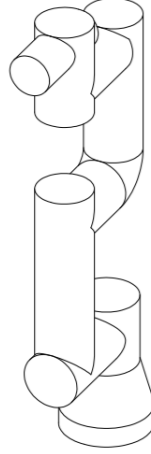


Figura 4.1: Boceto del brazo robótico propuesto

Cuadro 4.1: Parámetros del brazo robótico

Eslabones	1	2	3	4	5	6
Longitud	0.152	0.104	0.244	0.104	0.213	0.104
Masa	0	2	1	0.8	0.8	0.2
Centro de gravedad	[0, 0, 0]	[0, 0, 0]	[0, 0, 0]	[0, 0, 0]	[0, 0, 0]	[0, 0, 0]
Matriz de inercia	[0, 0, 0]	[0, 0, 0]	[0, 0, 0]	[0, 0, 0]	[0, 0, 0]	[0, 0, 0]
Fricción en eslabón	0.00148	0.00817	0.00138	7.12e-05	8.26e-05	3.67e-05
Fricción de Coulomb	0.395	0.126	0.132	0.0113	0.00926	0.00396
Inercia del motor	0.002	0.002	0.002	3.3e-05	3.3e-05	3.3e-05

Con estos datos claros, es posible empezar la realización de los modelos matemáticos.

4.1. Cinemática directa e inversa

4.1.1. Cinemática directa

La cinemática directa de un robot se refiere al cálculo de la posición y orientación del marco de referencia del efector final a partir de las coordenadas θ de sus articulaciones. [4]

4.1.1.1. Matriz de transformación homogénea

Según [4], existen tres usos principales para una matriz de transformación homogénea:

1. Para representar la configuración (posición y orientación) de un cuerpo rígido.
2. Para cambiar el marco de referencia en el cuál está representado un vector o un *frame*.
3. Para desplazar un vector o un *frame*.

Para el caso que nos ocupa, necesitamos la matriz de transformación homogénea desde la base fija del robot hasta su efector final, descrita con la ecuación siguiente:

$$\begin{aligned} {}^7_0T = & \mathcal{T}_z(a_1) \oplus \mathcal{R}_z(\theta_1) \oplus \mathcal{T}_x(b_1) \oplus \mathcal{R}_x(\theta_2) \oplus \mathcal{T}_z(c_1) \oplus \mathcal{R}_x(\theta_3) \\ & \oplus \mathcal{T}_z(d_1) \oplus \mathcal{T}_x(d_2) \oplus \mathcal{R}_x(\theta_4) \oplus \mathcal{T}_x(e_1) \oplus \mathcal{R}_z(\theta_5) \oplus \mathcal{T}_z(f_1) \oplus \mathcal{R}_x(\theta_6) \end{aligned} \quad (4.1)$$

En la imagen 4.2 podemos observar la cadena cinemática de nuestro brazo robótico, fue creada con un algoritmo en MATLAB con ayuda de la herramienta Robotic Toolbox, desarrollada por Peter Corke [5], dicho código puede consultarse en el Anexo 1.

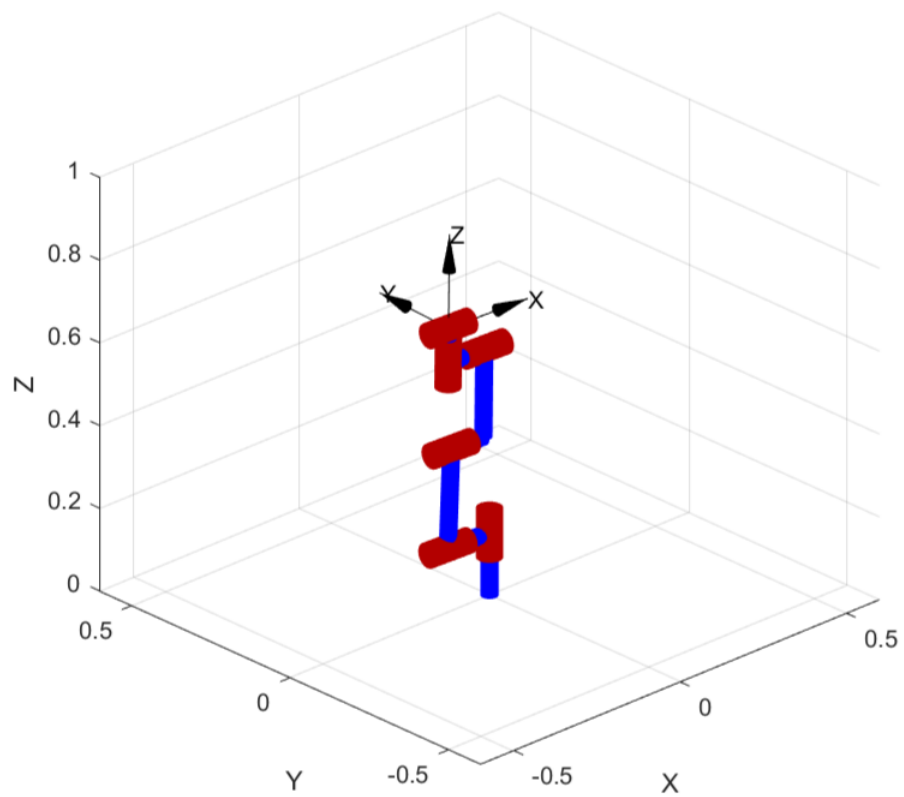


Figura 4.2: Cadena cinemática

4.1.1.2. Convención de Denavit-Hartenberg

En muchas aplicaciones, es necesario representar los parámetros cinemáticos de forma simplificada con ayuda de la notación Denavit-Hartenberg, con esta, podemos hacer uso de algoritmos de solución para encontrar los valores dinámicos, planeación de trayectorias, simulaciones, entre otras.

Peter Corke [6] propuso un acercamiento sencillo y sistemático para convertir una matriz de transformación homogénea como la de la ecuación 4.1 en los parámetros de Denavit-Hartenberg.

Se realizó dicho proceso con ayuda de Robotic Toolbox del mismo autor y se llegó a los resultados que se muestran en la tabla 4.2.

Cuadro 4.2: Parámetros Denavit Hartenberg

	θ [rad]	d [m]	a [m]	α [rad]
Articulación 1	θ_1	0.152	0	$\frac{\pi}{2}$
Articulación 2	θ_2	0	-0.244	0
Articulación 3	θ_3	0	-0.213	0
Articulación 4	θ_4	-0.012	0	$-\frac{\pi}{2}$
Articulación 5	θ_5	0.085	0	$\frac{\pi}{2}$
Articulación 6	θ_6	0	0	$-\frac{\pi}{2}$

4.1.2. Cinemática inversa

De manera parecida a como se manejó en la sección anterior, la librería de Robotic Toolbox tiene un método establecido para calcular la cinemática inversa dada una matriz de transformación homogénea.

En este texto se hará uso de la misma y no se abordará en mayor profundidad.

4.2. Cinemática de la velocidad

4.3. Modelo dinámico

Como se comentó al principio del capítulo, es necesario establecer una relación entre las posiciones, velocidades y aceleraciones deseadas y el torque o fuerzas que se debe ejercer en los actuadores, esto nos permitirá controlar el brazo robótico así como conocer los parámetros necesarios que los actuadores deben cumplir para satisfacer los requerimientos de velocidad y carga útil.

4.3.1. Formulación Newton-Euler

Para el desarrollo del modelo dinámico se favoreció la formulación de Newton-Euler en lugar de la formulación Lagrangiana, esto debido a su eficiencia computacional.

Para encontrar los torques a los que deben estar sometidos las articulaciones del brazo robótico se utilizará la herramienta que se ha utilizado a lo largo de este capítulo, con ayuda de ésta, se desarrolla un algoritmo en MATLAB donde se incluyen los parámetros necesarios para el modelo dinámico tales como masa, centro de gravedad, momento de inercia, fricción viscosa, fricción de Coulomb, inercia del motor y relación de engranes.

La lista de parámetros para cada articulación y eslabón fue descrita anteriormente en el cuadro 4.1.

Cabe destacar que algunos de los parámetros dinámicos son una aproximación optimista, al no tener aún un diseño final ni todos los componentes seleccionados, sin embargo, es funcional para una primera iteración del torque necesario para continuar con la selección de componentes.

Con los datos del cuadro 4.1 se selecciona los torques máximos en el escenario más demandante, esto es en una trayectoria en la cuál cada una de las articulaciones se somete a una mayor fuerza.

Con todos los datos correctamente particularizados, obtenemos una lista con todos los valores máximos, se puede apreciar en el cuadro 4.3.

Cuadro 4.3: Parámetros del brazo robótico

Articulaciones	1	2	3	4	5	6
Torque máximo	0.2	25.35	11.94	2.33	0.2	0.2

Capítulo 5

Componentes eléctricos

Aquí necesito seis motores, tres ODrive, seis encoders Una fuente de poder

Capítulo 6

Componentes mecánicos

Una vez determinado los componentes electrónicos necesarios para el correcto funcionamiento del brazo robótico es necesario determinar los componentes mecánicos.

6.1. Actuador

Una de las partes mecánicas más importantes es el actuador, es decir, el mecanismo que proporcionará la fuerza necesaria para mover las articulaciones del robot.

Para la correcta selección o desarrollo del actuador necesitamos dos parámetros que vimos en los capítulos anteriores, el primero es el torque necesario para realizar los desplazamientos del robot a la velocidades y aceleraciones requeridas y el segundo es la velocidad nominal del motor a utilizar.

Por norma general, los motores eléctricos y, para nuestro caso, los motores eléctricos sin escobillas, funcionan a velocidades altas y un torque relativamente pequeño, para esto es necesario utilizar reductores de velocidad, los cuales cumplen dos objetivos primordiales: reducir la velocidad en eje de salida y aumentar el torque en el mismo.

El siguiente paso es escoger un reductor de velocidad que cumpla con los requerimientos específicos de nuestro brazo robótico, los cuales son:

- Alta relación de transmisión, mayor a 10:1.

- Precisión, necesaria para realizar los movimientos deseados sin demasiado juego.
- Efectividad de transmisión, necesaria para no perder demasiada energía

Estos son algunos de los más importantes, existen otros factores a considerar como el tamaño final del reductor y la facilidad de manufactura.

Para el desarrollo de este brazo robótico se han considerado tres diferentes tipos de reductores de velocidad, cada uno con ventajas y desventajas que se mencionarán a continuación.

6.1.1. Engranajes planetarios

El primero en esta lista es un reductor con engranes planetarios,

Existen proyectos como el de [insertar cita después] OpenTorque Actuator de Gabrael Levine que utiliza engranes planetarios en el diseño de su actuador, logra una relación de transmisión de 8:1

En la imagen 6.1 podemos apreciar su diseño con engranes planetarios del tipo helicoidal así como una carcasa para el sistema de motor y actuador.

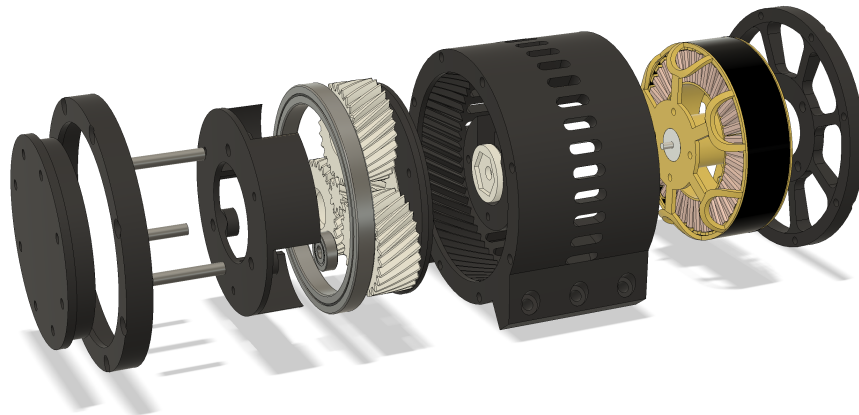


Figura 6.1: OpenTorque Actuator por Gabrael Levine

En esta propuesta la ventaja es la relativa facilidad de fabricación con tecnología de manufactura aditiva así como un tamaño y peso considerablemente pequeño.

Las desventajas son su relativamente pequeño relación de transmisión y su poca eficiencia de transmisión.

6.1.2. Engrane cicloidal

6.1.3. Harmonic Drive

Bibliografía

- [1] S. E. Zaatari, M. Marei, W. Li, and Z. Usman, “Cobot programming for collaborative industrial tasks: An overview,” *Robotics and Autonomous Systems*, vol. 116, pp. 162–180, jun 2019.
- [2] M. W. Spong, S. Hutchinson, and M. Vidyasagar, *Robot Modeling and Control*. WILEY, 2005.
- [3] W. K. Etienne Dombre, *Robot Manipulators: Modeling, Performance Analysis and Control (Control Systems, Robotics, and Manufacturing series)*. Wiley-ISTE, 2007.
- [4] K. M. N. University, I. Lynch, and F. C. S. N. U. Park, *Modern Robotics*. Cambridge University Press, 2017.
- [5] P. Corke, *Robotics, Vision and Control*. Springer International Publishing, 2017.
- [6] P. Corke, “A simple and systematic approach to assigning denavit–hartenberg parameters,” *IEEE Transactions on Robotics*, vol. 23, pp. 590–594, jun 2007.

Anexos

Anexo 1. Función de creación del brazo robótico

```
function roboticarm = myroboticarm(L1,L2,L3,L4,L5,L6,L7,m,r,  
    I1,I2,I3,I4,I5,I6,Jm,B,Tc,G,payload)  
%This function crea un objeto del tipo SerialLink  
% Con los parametros de mi brazo robotico  
  
s = 'Rz(q1) Tz(L1) Tx(L2) Rx(q2) Tz(L3) Rx(q3) Tx(L4) Tz(L5)  
    Rx(q4) Tx(L6) Rz(q5) Tz(L7) Rx(q6)';  
dh = DHFactor(s);  
roboticarm = eval(dh.command('roboticarm'))  
  
roboticarm.links(1,1).m = m(1);  
roboticarm.links(1,1).r = r(1,:);  
roboticarm.links(1,1).I = I1;  
roboticarm.links(1,1).Jm = Jm(1);  
roboticarm.links(1,1).B = B(1);  
roboticarm.links(1,1).Tc = Tc(1);  
roboticarm.links(1,1).G = G(1);  
  
roboticarm.links(1,2).m = m(2);  
roboticarm.links(1,2).r = r(2,:);
```

```

roboticarm.links(1,2).I = I2;
roboticarm.links(1,2).Jm = Jm(2);
roboticarm.links(1,2).B = B(2);
roboticarm.links(1,2).Tc = Tc(2);
roboticarm.links(1,2).G = G(2);

```

```

roboticarm.links(1,3).m = m(3);
roboticarm.links(1,3).r = r(3,:);
roboticarm.links(1,3).I = I3;
roboticarm.links(1,3).Jm = Jm(3);
roboticarm.links(1,3).B = B(3);
roboticarm.links(1,3).Tc = Tc(3);
roboticarm.links(1,3).G = G(3);

```

```

roboticarm.links(1,4).m = m(4);
roboticarm.links(1,4).r = r(4,:);
roboticarm.links(1,4).I = I4;
roboticarm.links(1,4).Jm = Jm(4);
roboticarm.links(1,4).B = B(4);
roboticarm.links(1,4).Tc = Tc(4);
roboticarm.links(1,4).G = G(4);

```

```

roboticarm.links(1,5).m = m(5);
roboticarm.links(1,5).r = r(5,:);
roboticarm.links(1,5).I = I5;
roboticarm.links(1,5).Jm = Jm(5);
roboticarm.links(1,5).B = B(5);
roboticarm.links(1,5).Tc = Tc(5);
roboticarm.links(1,5).G = G(5);

```

```

roboticarm.links(1,6).m = m(6);

```

```

roboticarm.links(1,6).r = r(6,:);
roboticarm.links(1,6).I = I6;
roboticarm.links(1,6).Jm = Jm(6);
roboticarm.links(1,6).B = B(6);
roboticarm.links(1,6).Tc = Tc(6);
roboticarm.links(1,6).G = G(6);

roboticarm.payload(payload, [0 0 0]);
roboticarm.base = SE3(0, 0, 0);

```

Anexo 2. Cinemática directa

```

% Universidad Veracruzana
% Tesis
% Angel Trujillo

clear;
clc;

L1 = 0.152;
L2 = -0.120;
L3 = 0.244;
L4 = 0.104;
L5 = 0.213;
L6 = -0.104;
L7 = 0.085;
qn = [0 0 0 0 0 0];
robotarm = myroboticarm(L1,L2,L3,L4,L5,L6,L7);
robotarm.fkine(qn)
robotarm.plot(qn)

```

Anexo 3. Cinemática inversa

```

% Universidad Veracruzana

```

```

% Tesis
% Angel Trujillo

clear;
clc;
L1 = 0.152;
L2 = -0.120;
L3 = 0.244;
L4 = 0.104;
L5 = 0.213;
L6 = -0.104;
L7 = 0.085;
qn = [0 0 0 0 0 0];
robotarm = myroboticarm(L1,L2,L3,L4,L5,L6,L7);
robotarm.plot(qn)
T = robotarm.fkine(qn)
qi = robotarm.ikine(T)

```

Anexo 4. Dinámica

```

% Universidad Veracruzana
% Tesis
% Angel Trujillo

clear;
clc;
%Medidas del brazo robotico
L1 = 0.152;
L2 = 0.104;
L3 = 0.244;
L4 = 0.104;
L5 = 0.213;

```

```

L6 = -0.104;
L7 = 0.085;
% Parametros dinamicos
m = [0 1 1 0.8 0.8 0.2]; % Vector de masas de los eslabones 1
    al 6.
r = [0 0 0; 0 0 0; 0 0 0; 0 0 0; 0 0 0; 0 0 0]; % Vector
    de centro de gravedad cada fila es un eslabon
I1 = [0 0 0; 0 0 0; 0 0 0];
I2 = [0 0 0; 0 0 0; 0 0 0];
I3 = [0 0 0; 0 0 0; 0 0 0];
I4 = [0 0 0; 0 0 0; 0 0 0];
I5 = [0 0 0; 0 0 0; 0 0 0];
I6 = [0 0 0; 0 0 0; 0 0 0];
Jm = [0.002 0.002 0.002 3.3e-05 3.3e-05 3.3e-05]; % Motor
    inertia
B = [0.00148 0.000817 0.00138 7.12e-05 8.26e-05 3.67e-05]; %
    Motor viscous friction
Tc = [0.395 0.126 0.132 0.0112 0.00926 0.00396]; %link
    coulomb friction
G = [1 1 1 1 1 1]; %Gear ratio
payload = 2;
qn = [0 0 0 0 0 0].*pi/180;
qz = [0 -90 0 0 -90 0].*pi/180;
q = jtraj(qz,qn,10);

robotarm = myroboticarm(L1,L2,L3,L4,L5,L6,L7,m,r,I1,I2,I3,I4,
    I5,I6,Jm,B,Tc,G,payload);

torque = robotarm.rne(q, 0*q, 0*q)
robotarm.plot(q, 'delay',1)

```