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

Índice de figuras

Índice de cuadros

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

Aquí empieza el viaje

1.1. Objetivos

El objetivo general de esta investigación consiste en diseñar y construir un brazo robótico 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. [1, 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

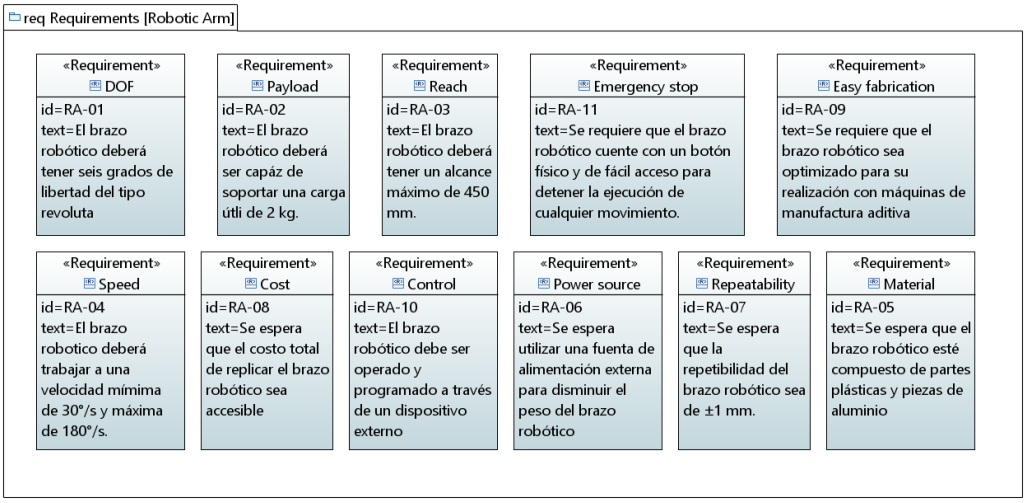

Figura 3.1: Diagrama de requerimientos

Capítulo 4

Modelo matemático

De acuerdo con [2, 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 estos modelos matemáticos, los cuales son necesarios para simular y predecir el comportamiento del mismo.

Para realizar estos modelos, es necesario contar con los parámetros físicos y geométricos del robot, los cuales, para una primera aproximación se mencionarán a continuación.

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.

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

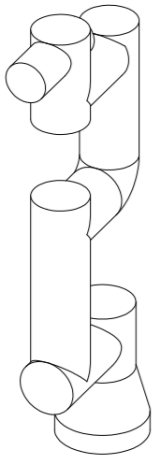

Figura 4.1: Boceto del brazo robótico propuesto

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 desde sus coordenadas θ. [3]

4.1.1.1. Matriz de transformación homogénea

Según [3], 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:

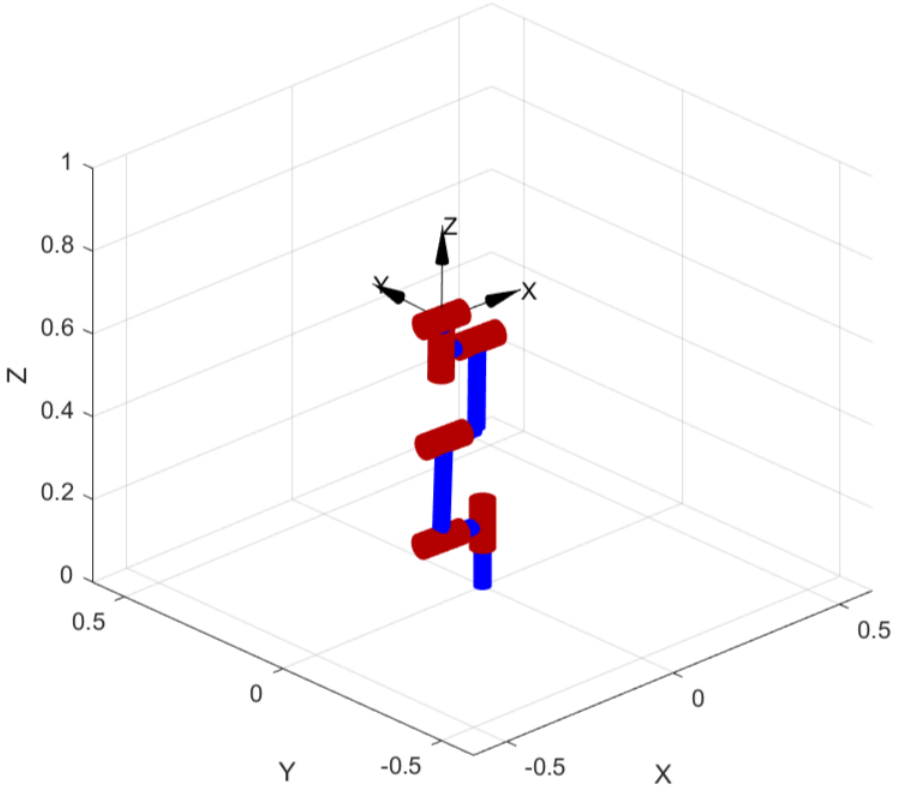

Figura 4.2: Cadena cinemática

$$
\begin{aligned}
{}^{7}_{0}T = \mathscr{T}_z(a_1) \oplus \mathscr{R}_z(\theta_1) \oplus \mathscr{T}_x(b_1) \oplus \mathscr{R}_x(\theta_2) \oplus \mathscr{T}_z(c_1) \oplus \mathscr{R}_x(\theta_3) \\
\oplus \mathscr{T}_z(d_1) \oplus \mathscr{T}_x(d_2) \oplus \mathscr{R}_x(\theta_4) \oplus \mathscr{T}_x(e_1) \oplus \mathscr{R}_z(\theta_5) \oplus \mathscr{T}_z(f_1) \oplus \mathscr{R}_x(\theta_6)
\end{aligned} \tag{4.1}
$$

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 [4], dicho código puede consultarse en el Anexo 1.

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 [5] propuso un acercamiento sencillo y sistemático para convertir una matriz de transformación homogenea 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 4.1.

Cuadro 4.1: 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 Lagraniana, esto debido a su eficiencia computacional.

Para encontrar los torques a los que deben estar sometidos las articulaciones del brazo robótico se utilizará la misma herramienta que se ha utilizado a lo largo de este capítulo, Robotic Toolbox de Peter Corke, se modifica el Anexo 1 para incluir los parámetros dinámicos 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 esta descrita en el cuadro 4.2.

Cuadro 4.2: Parámetros dinámicos

	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	$-\frac{\pi}{2}$

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 empezar la selección de componentes.

Con los datos del cuadro 4.2 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: Torque máximo por articulación

	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	$-\frac{\pi}{2}$

Capítulo 5

Componentes eléctricos

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

Bibliografía

[1] M. W. Spong, S. Hutchinson, and M. Vidyasagar, *Robot Modeling and Control*. WILEY, 2005.

[2] W. K. Etienne Dombre, *Robot Manipulators: Modeling, Performance Analysis and Control (Control Systems, Robotics, and Manufacturing series)*. Wiley-ISTE, 2007.

[3] K. M. N. University, I. Lynch, and F. C. S. N. U. Park, *Modern Robotics*. Cambridge University Press, 2017.

[4] P. Corke, *Robotics, Vision and Control*. Springer International Publishing, 2017.

[5] 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 2 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 0.3 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 3.67
   e-05];  %  Motor viscous friction
Tc = [0.395 0.126 0.132 0.0112 0.00926 0.00396]; %
   link coulomb friction
G = [10 10 4 4 2 2]; %Gear ratio
payload = 2;
qn = [0 0 0 0 0 0].*(pi/180);
```

```
qz = [0 90 0 0 0 0].*(pi/180);


q = jtraj(qz,qn,20);

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.teach
```