



Universidad Veracruzana

Facultad de Ingeniería de la construcción y el habitat

TESIS:

**Diseño de un brazo robótico colaborativo de seis
grados de libertad**

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. Planteamiento del problema	9
1.2. Objetivos	9
1.3. Justificación	10
1.4. Hipótesis	11
2. Marco teórico	12
2.1. Brazos robóticos industriales	12
2.1.1. Clasificación de los robots	12
2.2. Robótica colaborativa	13
2.3. Sistemas de manufactura flexible	16
2.4. Ingeniería de sistemas basado en modelos	16
3. Metodología	17
3.1. Modelos matemáticos	17

3.1.1.	Cinemática directa e inversa.	19
3.1.1.1.	Cinemática directa	19
3.1.1.2.	Cinemática inversa	22
3.1.2.	Cinemática de la velocidad	23
3.1.3.	Modelo dinámico	23
3.1.3.1.	Formulación Newton-Euler	24
3.2.	Componentes electrónicos	25
3.3.	Componentes mecánicos	25
3.4.	Actuador	25
3.4.1.	Engranajes planetarios	26
3.4.2.	Engrane cicloidal	26
3.4.3.	Harmonic Drive	26
Bibliografía		29
Anexos		30
Anexo 1.	Función de creación del brazo robótico	30
Anexo 2.	Cinemática directa	32
Anexo 3.	Cinemática inversa	32
Anexo 4.	Dinámica	33

Índice de figuras

2.1. Escenarios colaborativos	14
2.2. Robot colaborativo UR5 de Universal Robots	15
3.1. Boceto del brazo robótico propuesto	18
3.2. Cinemática directa de un brazo robótico de 3 grados de libertad [1]	20
3.3. Cadena cinemática	21
3.4. OpenTorque Actuator por Gabrael Levine	27

Índice de cuadros

3.1. Parámetros del brazo robótico	18
3.2. Parámetros Denavit Hartenberg	22
3.3. Parámetros del brazo robótico	25

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.

...

You always can find the latest version of this file in: <http://github.com/Trujillo19/Thesis>.

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. Las desventajas consisten en sacrificar la carga útil máxima así como la eficiencia y el alcance de los robots industriales tradicionales. En resumen, los robots colaborativos son un excelente compromiso entre el trabajo manual y la automatización industrial. [2]

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 desarrollado productos para este nicho de mercado. Por su parte, las grandes empresas de manufactura como Audi, Volkswagen y Nissan han incorporado robots colaborativos en sus líneas de ensamblaje. [2]

Aún con el crecimiento en la demanda de los robots colaborativos, el precio de éstos no ha sufrido grandes cambios desde su lanzamiento, pues su precio promedio ronda en los **\$50,000 dólares**, lo cual es prohibitivo para compañías en países emergentes.

El propósito de esta investigación consiste en desarrollar un brazo robótico colaborativo de coste accesible para que las pequeñas y medianas empresas puedan aumentar la eficiencia y calidad en sus procesos así como librar a sus trabajadores de tareas repetitivas

o potencialmente peligrosas.

1.1. Planteamiento del problema

En el campo de la robótica y la automatización existe una brecha pronunciada entre la automatización industrial disponible para las grandes empresas de manufactura y lo que las pequeñas y medianas empresas se pueden costear.

Si bien es cierto que la demanda de brazos robóticos en la industria tiene una tendencia en aumento en todo el mundo [3], las innovaciones en este campo particular han sido pocas.

Recientemente, un nuevo tipo de brazos robóticos han llegado al mercado, denominados *cobots* o robots colaborativos, los cuales, tienen como función principal el poder trabajar lado a lado con humanos sin poner en riesgo la seguridad de ambos.

Al ser un desarrollo nuevo, el precio de los brazos robóticos colaborativos es prohibitivo para empresas pequeñas, por lo que en este documento se trata de desarrollar un brazo robótico colaborativo de código abierto y de coste accesible.

1.2. Objetivos

El objetivo general de esta investigación consiste en diseñar un brazo robótico colaborativo de seis grados de libertad. Los objetivos específicos se mencionan a continuación:

- Desarrollar el modelo matemático cinemático y dinámico de un brazo robótico de seis grados de libertad.
- Diseñar un brazo robótico de coste accesible y fácil reproducción.
- Optimizar el modelado de piezas para su correcta manufactura con máquinas de manufactura aditiva.
- Utilizar una metodología apegada a la ingeniería de sistemas basadas en modelos.
- Compartir su diseño, componentes y software con una licencia de código abierto.

1.3. Justificación

El desarrollo de un brazo robótico colaborativo de seis grados de libertad que se pretende realizar es relevante en diversos ámbitos del conocimiento, tales como la robótica, la inteligencia artificial y la metodología de ingeniería de sistemas basado en modelos.

Para lograr la descentralización deseada en la industria 4.0 es necesario que los medios de producción sean democráticos y estén al alcance de la mayor cantidad de personas posibles, es por esto que es necesario desarrollar tecnología de coste accesible y de código abierto.

El elevado precio de los robots colaborativos actuales se basa en diferentes factores, entre los que se encuentran:

- Certificaciones de seguridad, resistencia contra los elementos o seguridad ambiental.
- Accesorios incluidos tales como la consola de control y aprendizaje o elementos de anclaje.
- Elementos protegidos por patentes o excesivamente costosos, como los *harmonic drive*.
- Investigación y desarrollo.

Es posible eliminar la mayoría de estos costos al desarrollar únicamente una plataforma mecánica y de software robusta sobre la cual la comunidad pueda desarrollar, remplazando los elementos costosos por alternativas más baratas y eliminando los accesorios y las certificaciones.

Otro punto importante a tener en cuenta, es que la naturaleza de este proyecto es servir como plataforma para el desarrollo de la investigación, pues con ayuda del brazo robótico se podrá desarrollar investigación en el área de inteligencia artificial, tales como aprendizaje reforzado o visión computacional.

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

1.4. Hipótesis

Es posible realizar un brazo robótico colaborativo de coste accesible para implementarlo en sistemas de manufactura flexible. Así mismo, es posible realizar la mayoría de los componentes estructurales por medio de manufactura aditiva, en especial los reductores de velocidad necesarios para alcanzar los requisitos de carga útil máxima y velocidad de operación.

Capítulo 2

Marco teórico

2.1. Brazos robóticos industriales

Antes de entrar en el tema principal de este documento, es importante conocer el antecedente o lo que se trata de mejorar, en nuestro caso, es imposible hablar de robots colaborativos sin primero abordar los brazos robóticos industriales.

Un brazo robótico es un manipulador, usualmente programable con funciones similares a las de un brazo humano. Los eslabones de este manipulador estan conectados por articulaciones que permiten el movimiento rotacional o translacional. [4] [5]

De acuerdo con [6], 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.

La Federación Internacional de Robótica ha reportado un crecimiento promedio anual a partir del 2013 del 19 % en la demanda de brazos robóticos, siendo la industria automotriz la dominante, seguida por la industria eléctrica y electrónica. [3]

2.1.1. Clasificación de los robots

La clasificación de los brazos robóticos tiene dos grandes ramas, una de estas es por la fuente de su energía y la otra por la geometría de trabajo.

La fuente de energía puede ser eléctrica o hidráulica, la mayoría de los manipuladores hoy en día usan servomotores o motores a pasos de corriente continua [5].

Dependiendo de la geometría de trabajo, los brazos robóticos se dividen en: cartesianos, cilíndricos, esféricos, SCARA o articulados.

2.2. Robótica colaborativa

La robótica colaborativa surge como una rama de la robótica dispuesta a hacer más fácil el trabajo en líneas de producción, así como aliviar problemas en la espalda relacionados con tareas de ensamblado final en posiciones no ergonómicas [7][8].

Los robots colaborativos han sido desarrollados para trabajar de manera segura al lado de humanos, tomando en consideración la seguridad del operador, la del mismo robot y la correcta realización de la tarea a realizar. Por esto, están equipados con medidas de seguridad para reconocer el ambiente en el que trabajan, tales como sistemas de evasión de obstáculos y detección de colisiones, implementadas a través de sensores o de mecanismos de compilación pasiva.

Escenarios de colaboración. Es claro que no todas las aplicaciones donde se implemente un robot colaborativo requieren las mismas consideraciones, es por esto que existe una clasificación de acuerdo al grado de interacción con el operador y la dependencia entre éste y el robot colaborativo. Esos escenarios se mencionan a continuación y se ilustran en la figura 2.1 [2].

- **Independiente.** Un operador y un robot colaborativo trabajan en distintas piezas de trabajo, cada uno con un proceso de manufactura individual. El elemento colaborativo es debido a la co-presencia en la misma área de trabajo.
- **Simultanea.** Un operador y un robot colaborativo operan en procesos separados en la misma pieza de trabajo al mismo tiempo. No existe una dependencia de tarea entre ellos, sin embargo, el cobot debe estar conciente y respetar el espacio del operador.

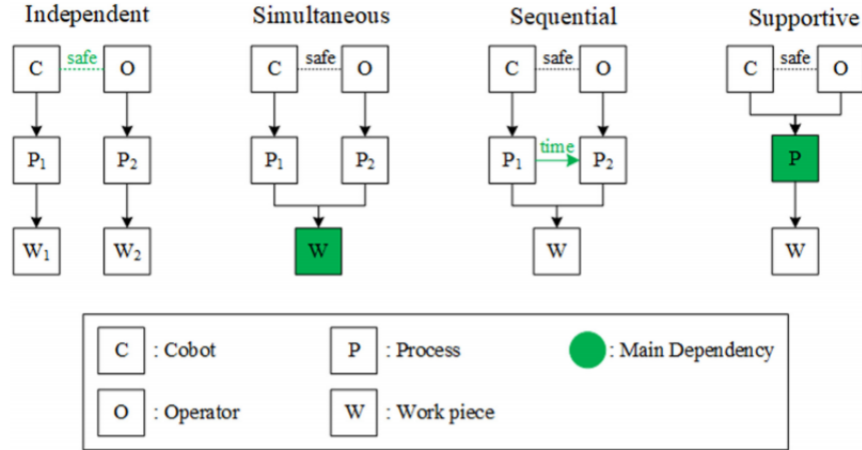


Figura 2.1: Escenarios colaborativos

- **Secuencial.** Un operador y un cobot relizan un proceso de manufactura secuencial en la misma pieza de trabajo. Existe dependencia de tiempo entre ambos a lo largo del proceso.
- **Soporte.** Un operador y un robot trabajan en el mismo proceso de manera interactiva. Existe una dependencia entre las acciones del robot y el operador, es decir, sin uno de ellos, la tarea no podría realizarse.

Con esto en mente, es fácil deducir que la mayoría de las investigaciones en el área de robótica colaborativa están encaminadas a estudiar o mejorar la detección de colisiones para salvaguardar la integridad de las personas que deberán trabajar junto a los cobots.

Incluso antes de que se acuñara el término cobots, ya había investigaciones encaminadas a la evasión de obstáculos [9], dónde se plantea la solución a través de algoritmos de planeación de ruta.

En la mayoría de los robots colaborativos que existen en el mercado hoy en día se utilizan sensores de fuerza-torque en cada una de las articulaciones, sin embargo, los sensores montados en el robot colaborativo no son la única forma de proveer la seguridad necesaria, en [10], se aborda el uso de sensores de visión computacional a través del dispositivo Kinect V2 de Microsoft para modificar la trayectoria y así evadir obstáculos.

En [11], se concluye que es posible detectar colisiones utilizando sensores de fuerza-torque únicamente en la base y muñeca de un brazo robótico. Los sensores de fuerza-torque

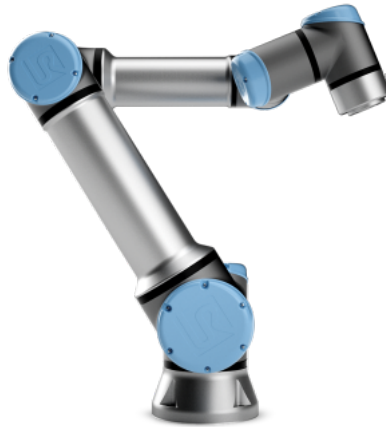


Figura 2.2: Robot colaborativo UR5 de Universal Robots

son sumamente caros, por esta razón existen investigaciones como [12] donde se trabaja en desarrollar nuevos sensores de fuerza-torque con gran velocidad de respuesta y bajo costo.

De igual forma, se han desarrollado estudios sobre como es posible desarrollar robots colaborativos sin la necesidad de sensores de fuerza-torque.

En [13], se propone un método de detección de colisión basado en la ley de control no-lineal adaptativo, donde se utiliza la diferencia entre el torque obtenido realmente en contra del calculado basado en el modelo dinámico.

Por otra parte, en [14] también se propone el desarrollo de un algoritmo de detección de colisión basado en el modelo dinámico del robot, en este se utiliza la corriente suministrada al motor y la información que proporcionen los sensores de posición para medir discrepancias y detectar colisiones, lo que reduce el costo dedicado a sensores que requiere un robot colaborativo. En esta investigación se tratará de replicar esta metodología para lograr un algoritmo de detección de colisión.

Uno de los ejemplos más populares de robots colaborativos es la línea de robots UR creada por Universal Robots, quién ha vendido más de 39,000 unidades de estos cobots. Una imagen del mismo puede ser apreciada en la figura 2.2.

2.3. Sistemas de manufactura flexible

2.4. Ingeniería de sistemas basado en modelos

Capítulo 3

Metodología

3.1. Modelos matemáticos

De acuerdo con [15], 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, para una primera aproximación, se utilizarán valores experimentales. Éstos se mencionan en la tabla 3.1.

Otros parámetros necesarios para el desarrollo del modelo matemático del brazo robótico son el alcance total del brazo, el cual deberá ser de mínimo 500 mm, la velocidad, la

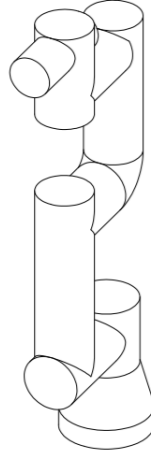


Figura 3.1: Boceto del brazo robótico propuesto

Cuadro 3.1: 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

cuál deberá estar en un rango entre 5 RPM y 30 RPM, y por último, la carga útil, la cuál deberá ser de 2 kg.

En la figura 3.1 podemos ver un boceto del brazo robótico que se planea implementar.

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

3.1.1. Cinemática directa e inversa.

En esta sección se desarrollará el primer modelo matemático del brazo robótico, la cinemática directa e inversa, la cual se trata de encontrar la posición final del brazo dependiendo de las coordenadas θ de sus articulaciones o, para el caso de la cinemática inversa, calcular las coordenadas θ necesarias para llegar a una posición (x, y, z) dada.

3.1.1.1. Cinemática directa

Como ya se mencionó anteriormente, la cinemática directa de un brazo robótico 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. [1]

Un ejemplo de esto lo podemos encontrar en la figura 3.2, donde se analiza la cinemática directa de un brazo robótico de tres grados de libertad. En este ejemplo podemos apreciar que la longitud de los eslabones se representan como L_1, L_2, L_3 respectivamente y se ha escogido un marco de referencia fijo denominado $\{0\}$, mientras que el marco de referencia del efector final es $\{4\}$. [1]

Entonces, para conocer la posición final es necesaria la siguiente ecuación:

$$x = L_1 \cos(\theta_1) + L_2 \cos(\theta_1 + \theta_2) + L_3 \cos(\theta_1 + \theta_2 + \theta_3) \quad (3.1)$$

$$y = L_1 \sin(\theta_1) + L_2 \sin(\theta_1 + \theta_2) + L_3 \sin(\theta_1 + \theta_2 + \theta_3) \quad (3.2)$$

$$\phi = \theta_1 + \theta_2 + \theta_3 \quad (3.3)$$

Las ecuaciones 3.1, 3.2 y 3.3 se les conoce como **ecuación de cinemática directa** y aunque es válida, se vuelve rápidamente compleja para un brazo robótico de seis grados de libertad, por esta razón, el enfoque sistemático de resolución es utilizar matrices de transformación homogénea y la convención de Denavit-Hartenberg.

Matriz de transformación homogénea. Una matriz de transformación homogénea no es nada más que la representación matricial de un movimiento rígido [6], así, la posición y orientación del efector final puede ser fácilmente calculada utilizando multiplicación de matrices.

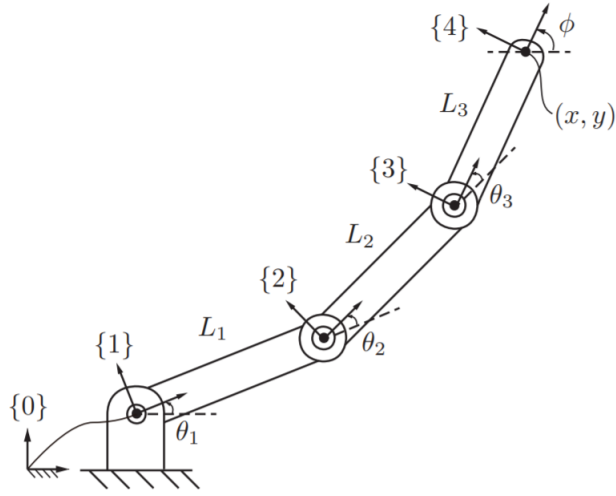


Figura 3.2: Cinemática directa de un brazo robótico de 3 grados de libertad [1]

La forma más general de una matriz de transformación homogénea puede ser escrita de la siguiente manera:

$$T_1^0 = \begin{bmatrix} n_x & s_x & a_x & d_x \\ n_y & s_y & a_y & d_y \\ n_z & s_z & a_z & d_z \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} n & s & a & d \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.4)$$

En dónde n es un vector representando la dirección x_1 en el sistema $o_0x_0y_0z_0$, de manera similar, s representa la dirección de y_1 y a representa la dirección de z_1 . Por su parte, d es un vector representando la posición desde el origen o_0 hasta el origen o_1 en el marco de referencia $o_0x_0y_0z_0$.

Según [1], 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 u otro marco de referencia.
3. Para desplazar un vector o un marco de referencia.

Para el caso que nos ocupa, necesitamos la matriz de transformación homogénea desde

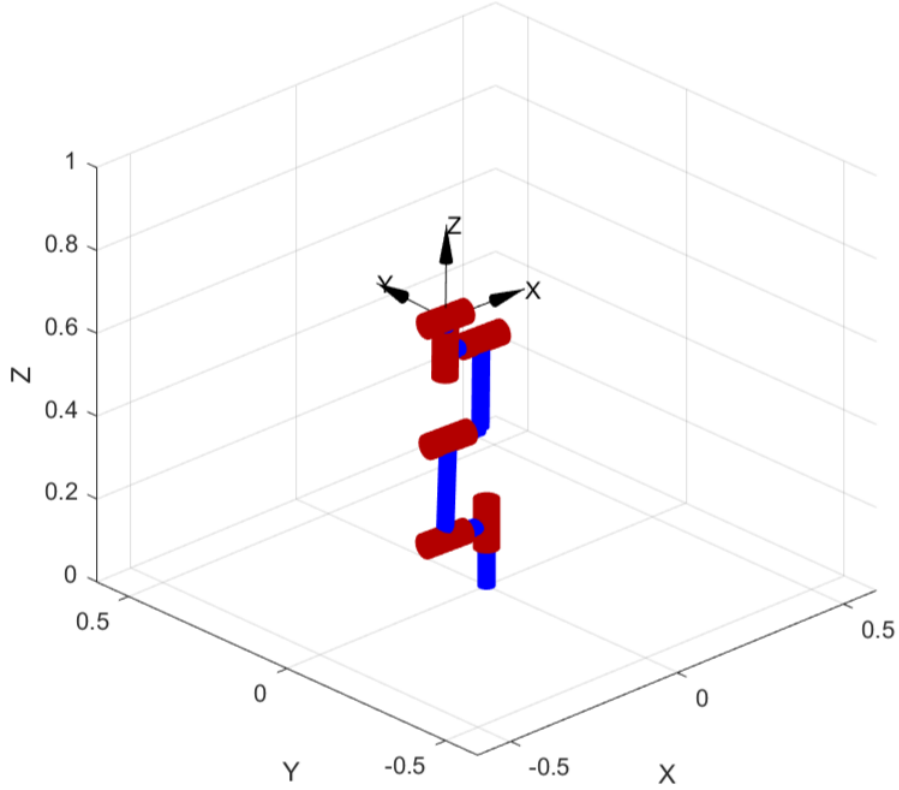


Figura 3.3: Cadena cinemática

la base fija del robot hasta su efector final, descrita con la ecuación siguiente:

$$\begin{aligned}
 {}^7_0T = & \mathcal{I}_z(a_1) \oplus \mathcal{R}_z(\theta_1) \oplus \mathcal{I}_x(b_1) \oplus \mathcal{R}_x(\theta_2) \oplus \mathcal{I}_z(c_1) \oplus \mathcal{R}_x(\theta_3) \\
 & \oplus \mathcal{I}_z(d_1) \oplus \mathcal{I}_x(d_2) \oplus \mathcal{R}_x(\theta_4) \oplus \mathcal{I}_x(e_1) \oplus \mathcal{R}_z(\theta_5) \oplus \mathcal{I}_z(f_1) \oplus \mathcal{R}_x(\theta_6)
 \end{aligned} \tag{3.5}$$

Dónde $a_1, b_1, c_1, d_1, d_2, e_1$ y f_1 son las longitudes de los eslabones del brazo robótico.

En la imagen 3.3 podemos observar la cadena cinemática de nuestro brazo robótico, fue creada con un algoritmo en MATLAB con ayuda de la herramienta Robotic Toolbox [16], dicho código puede consultarse en el Anexo 1.

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 convención se simplifica considerablemente el análisis.

Con esta notación podemos hacer uso de algoritmos de solución para encontrar los valores dinámicos, planeación de trayectorias y simulaciones, entre otros valores.

Cuadro 3.2: 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	$-\frac{\pi}{2}$

En [17], se propone un acercamiento sencillo y sistemático para convertir una matriz de transformación homogénea como la de la ecuación 3.5 en los parámetros de Denavit-Hartenberg.

Al realizar dicho proceso se llegó a los resultados que se muestran en la tabla 3.2.

Con esto, tenemos una vista simplificada de nuestro brazo robótico, que describe su cinemática directa usando únicamente cuatro parámetros por articulación. El algoritmo desarrollado puede consultarse en el Anexo 2.

3.1.1.2. Cinemática inversa

Una vez terminada la cinemática directa, toca al turno de abordar la cinemática inversa. Hasta este punto podemos calcular la posición del efector final utilizando las variables de las articulaciones, ahora, procederemos a encontrar las variables de las articulaciones dada una posición y orientación del efector final.

Para nuestro caso, un brazo robótico de seis grados de libertad, la solución tiene doce ecuaciones con seis incógnitas, sin embargo, nueve ecuaciones se derivan de la matriz de rotación dentro de la matriz de transformación homogénea 6_0T , esto deja únicamente tres ecuaciones independientes, agregando las tres ecuaciones del vector de posición dentro de la matriz de transformación homogénea 6_0T nos da como resultado seis ecuaciones con seis incógnitas.

Estas ecuaciones son no-lineales y transcendentales, lo cual hace más difícil encontrar una solución y como cualquier conjunto de ecuaciones no-lineales, puede existir una sola

solución o múltiples soluciones. [18]

Para resolver el problema de encontrar la cinemática inversa existen dos métodos posibles: *la solución de forma cerrada* y la solución numérica.

El consenso de la mayoría de los autores, tales como [6] y [18] es que se prefiere encontrar la solución de forma cerrada por dos razones principales, la velocidad de solución y la forma en la que se encuentra una solución, es decir, como la cinemática inversa puede tener muchas soluciones se pueden desarrollar reglas para favorecer un tipo de solución con respecto de otra.

Existen algunos artículos como [19] en el cuál se ejemplifica la manera de resolver la cinemática inversa de un robot manipulador de seis grados de libertad de forma analítica.

Queda claro que resolver la cinemática inversa no es tarea sencilla, por esta razón y de manera parecida a como se manejó en la sección anterior, se utilizará la librería *Robotic Toolbox*, la cual incluye un método establecido para calcular la cinemática inversa dada una matriz de transformación homogénea. El algoritmo desarrollado puede consultarse en el Anexo 3.

Así mismo, es importante notar que esta solución se utilizará únicamente en la etapa de desarrollo del brazo robótico y se tendrá que implementar un algoritmo de solución de cinemática inversa en el control final del robot.

3.1.2. Cinemática de la velocidad

Soon.

3.1.3. Modelo dinámico

Hasta ahora, se ha descrito el movimiento del robot sin consideración de las fuerzas y los torques necesarios para producir dicho movimiento, ahora toca el turno de analizarlas.

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.

Para lograrlo, existen dos formulaciones que se pueden seguir, la formulación Euler-Lagrange o la formulación Newton-Euler. Ambos métodos llegarían a la misma respuesta, sin embargo el camino será diferente.

En la formulación Euler-Lagrange se trata al robot como un todo y se realiza el análisis utilizando la formulación Lagrangiana (la diferencia entre energía cinética y energía potencial), por su parte, la formulación Newton-Euler divide el manipulador en cada uno de sus eslabones y describe las ecuaciones que definen su movimiento lineal y angular. [6]

En esta parte de la investigación y como parte de los modelos matemáticos necesarios para seguir diseñando el robot, se utilizará la *formulación Newton-Euler* para obtener los torques y fuerzas necesarias.

3.1.3.1. Formulación Newton-Euler

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 ella, 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 3.1.

Cabe destacar que algunos de los parámetros dinámicos son una aproximación optimista, esto debido a 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 3.1 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.

Cuadro 3.3: 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

Con todos los datos correctamente particularizados, obtenemos una lista con todos los valores máximos, se puede apreciar en el cuadro 3.3.

El algoritmo realizado está disponible para consulta en el Anexo 4. Así como el repositorio de esta tesis.

3.2. Componentes electrónicos

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

3.3. Componentes mecánicos

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

3.3.1. Actuadores

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.

3.3.1.1. Engranés 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 3.4 podemos apreciar su diseño con engranes planetarios del tipo helicoidal así como una carcasa para el sistema de motor y actuador.

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.

3.3.1.2. Engrane cicloidal

3.3.1.3. Harmonic Drive

Bibliografía

- [1] K. M. N. University, I. Lynch, and F. C. S. N. U. Park, *Modern Robotics*. Cambridge University Press, 2017.
- [2] 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.
- [3] I. F. of Robotics, *Executive Summary World Robotics 2019 Industrial Robots*. International Federation of Robotics, 2019.
- [4] R. Gautam, A. Gedam, A. Zade, and A. Mahawadiwar, “Review on development of industrial robotic arm,” *International Research Journal of Engineering and Technology*, 2017.
- [5] R. J. Schilling, *Fundamentals of Robotics : Analysis and Control*. Prentice Hall of India, 2001.
- [6] M. W. Spong, S. Hutchinson, and M. Vidyasagar, *Robot Modeling and Control*. WILEY, 2005.
- [7] I. F. of Robots, “Demistifying collaborative robots,” *IFR*, 2018.
- [8] M. A. Peshkin, “Cobot architecture,” *IEEE TRANSACTIONS ON ROBOTICS AND AUTOMATION*, 2001.
- [9] O. Khatib, “Real-time obstacle avoidance for manipulators and mobile robots,” in *Autonomous Robot Vehicles*, pp. 396–404, Springer New York, 1986.

- [10] B. Teke, M. Lanz, J.-K. Kamarainen, and A. Hietanen, “Real-time and robust collaborative robot motion control with microsoft kinect $\text{\textcircled{R}}$ v2,” in *2018 14th IEEE/ASME International Conference on Mechatronic and Embedded Systems and Applications (MESA)*, IEEE, jul 2018.
- [11] S. Lu, J. Chung, and S. Velinsky, “Human-robot collision detection and identification based on wrist and base force/torque sensors,” in *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, IEEE, 2005.
- [12] T.-P. Phan, P. C.-P. Chao, J.-J. Cai, Y.-J. Wang, S.-C. Wang, and K. Wong, “A novel 6-DOF force/torque sensor for COBOTs and its calibration method,” in *2018 IEEE International Conference on Applied System Invention (ICASI)*, IEEE, apr 2018.
- [13] T. Matsumoto and K. Kosuge, “Collision detection of manipulator based on adaptive control law,” in *2001 IEEE/ASME International Conference on Advanced Intelligent Mechatronics. Proceedings (Cat. No.01TH8556)*, IEEE, 2001.
- [14] S. Chen, M. Luo, and F. He, “A universal algorithm for sensorless collision detection of robot actuator faults,” *Advances in Mechanical Engineering*, vol. 10, p. 168781401774071, jan 2018.
- [15] W. K. Etienne Dombre, *Robot Manipulators: Modeling, Performance Analysis and Control (Control Systems, Robotics, and Manufacturing series)*. Wiley-ISTE, 2007.
- [16] P. Corke, *Robotics, Vision and Control*. Springer International Publishing, 2017.
- [17] P. Corke, “A simple and systematic approach to assigning denavit–hartenberg parameters,” *IEEE Transactions on Robotics*, vol. 23, pp. 590–594, jun 2007.
- [18] J. J. Craig, *Introduction to Robotics: Pearson New International Edition*. Pearson HigherEducation, 2013.
- [19] S. Chen, M. Luo, O. Abdelaziz, and G. Jiang, “A general analytical algorithm for collaborative robot (cobot) with 6 degree of freedom (DOF),” in *2017 International Conference on Applied System Innovation (ICASI)*, IEEE, may 2017.

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)

```