



Proyecto Final

Introducción

Hoy en día los brazos robóticos se utilizan a menudo para realizar tareas complejas o muy repetitivas con el fin de eliminar el tipo de errores que una persona cansada cometería después de hacer una tarea tediosa por largos periodos. Los avances tecnológicos han permitido que dispositivos con una amplia gama de sensores sean ideados y que evidentemente estén en todo nuestro entorno.

Es aquí donde entra nuestro interés por fabricar un modelo funcional de un brazo robótico a partir de cálculos realizados por nosotros para realizar dicho modelo.

Objetivos

Diseñar y prototipar un brazo robótico con 4 grados de libertad y un efector final.

Realizar los análisis de los elementos de dicho brazo robótico para caracterizar el diseño por medio de sensores, actuadores y un material previamente seleccionado.

Obtener el modelado cinemático directo de los eslabones de dicho brazo robótico.

Fabricar en el mundo real el modelo diseñado y cumplir con un conjunto de pruebas básicas.

Planteamiento del Problema

Se presenta la relevancia y contextualización de la cinemática de un sistema de brazo robótico con matriz de Denavit-Hartenberg que permite diseñar de manera concreta el movimiento que se desea en un robot cinemático. Se comprenden tanto los movimientos de rotación y traslación al igual que sus movimientos angulares, el presente proyecto a través de la relación de las matrices con el fenómeno físico del movimiento se plantea la premisa de resolver un conjunto de movimiento dentro del sistema robótico para controlar de manera concreta el modelo físico del brazo robótico con 5 grados de libertad.

Materiales

Para la parte electrónica de nuestro brazo robótico tenemos pensado lo siguiente para cubrir el brazo (funcionamiento) y un control con el cual podremos controlarlo.

- Arduino Uno o Arduino Mega 2560 (1).
- Servomotor metálico TowerPro MG90S (4).
- Micro Servomotor metálico TowerPro MG90 (2).
- Micro Switch Push Button 4 Pines (8).
- Interruptor de 5V (1).
- Resistencia de 10 K Ω (8).
- Resistencia de 220 Ω (1).
- Led verde (1).



- Protoboard 1 bloque, 2 tiras, 400 puntos (1).
- Cargador de Carga Rápida 5V, hasta 2A. (1)
- Cable USB-MicroUSB (1).

Para la parte del cuerpo se está contemplando el uso de MDF, cartoncillo o dependiendo del costo imprimir las piezas mediante impresión 3D.

Sobre el servomotor y cálculo de potencias:

- Peso: 13.4 g
- Dimensiones: 22.5 x 12 x 35.5
- torque: 1.8 kgf·cm (4.8V)
- Velocidad de operación: 0.1 s/60 degree
- Voltaje de operación: 4.8 V
- Ancho de banda: 5 μ s

Marco teórico

Un robot es un dispositivo mecánico que posee articulaciones móviles enfocadas en la manipulación, estos dispositivos desempeñan actividades bajo la supervisión humana directa o con software previamente definido.

Robots Manipuladores

Un robot articulado es un manipulador reprogramable multifuncional y es diseñado con el fin de mover materiales, piezas o dispositivos especializados a través de movimientos programados variables para la realización de una diversidad de tareas.

Para ello se deben definir los grados de libertad del manipularos, que no son más que los movimientos independientes que puede realizar el robot (los movimientos posibles).

Otro aspecto importante que considerar es la capacidad de carga que tiene el robot para manipular objetos, estos son los pares necesarios para mover la carga los cuales varían según la configuración que el robot tenga, esta carga manipulable por todo el volumen manipulado se conoce como “Carga Útil”

Por último, se debe definir el volumen de trabajo del manipulador el cual es el espacio dentro del que se puede desplazar el punto donde se ensamble el efector final.

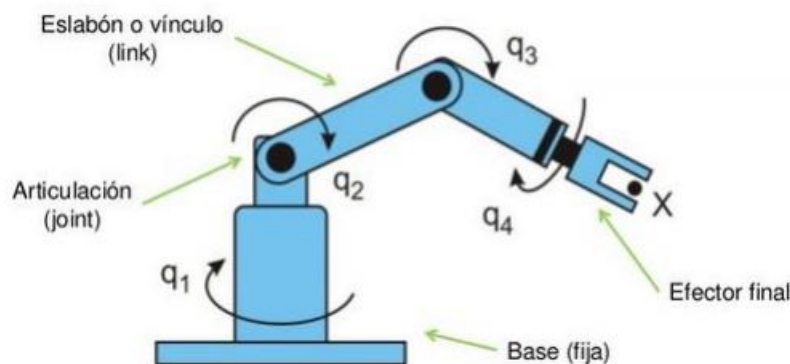


Figura 1.0 “Estructura básica de un manipulador”



Configuración de los robots manipuladores

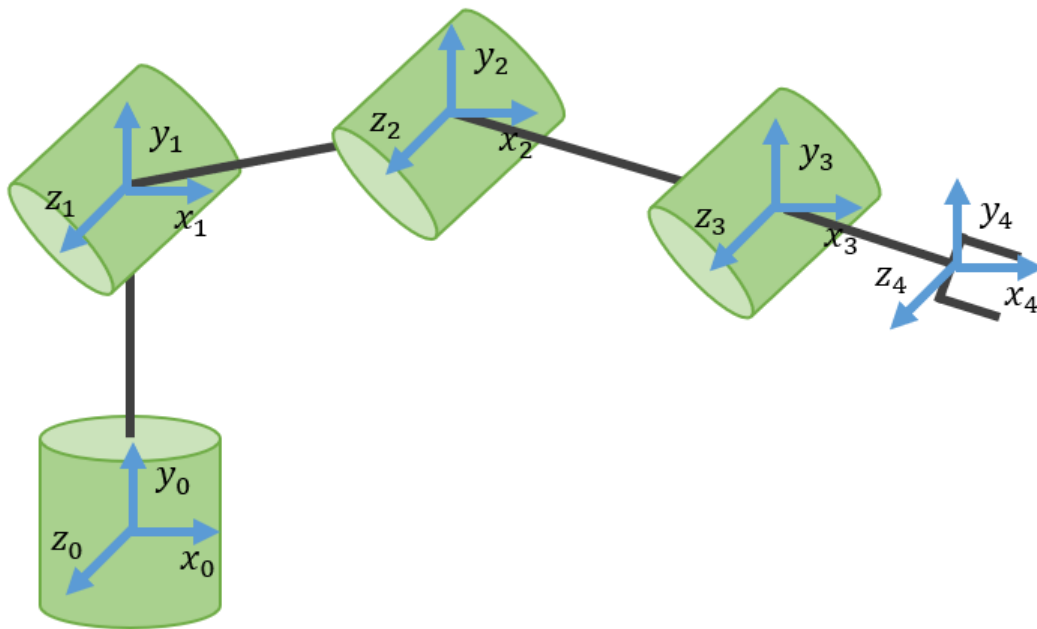
La configuración de un robot manipulador se define a partir del tipo de articulación que éste tenga, sin embargo, la más usada es la articulación de rotación debido a que suministra solo un grado de rotación consistente alrededor de un eje de articulación.

En nuestro caso utilizaremos éste tipo de articulaciones debido a su simpleza en cálculo.

Modelo

El brazo robótico planteado tendrá 5 grados de libertad que sea capaz de manipular objetos en situaciones que puedan comprometer la seguridad de una persona o supervisión de procesos a distancia.

En nuestro caso hicimos un robot cuyas articulaciones sean de tipo rotacional y el resultado fue el siguiente:



Cálculo de matriz de Denavit–Hartenberg

	Θ	d	a	α
H1	Q1	D1	0	-90
H2	Q2	D2	0	+90
H3	Q3	D3	0	0
H4	Q4	D4	0	0
H5	Q5	D5	0	-90

H1:



$$\begin{matrix} \cos(q_1) & -\sin(q_1) & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & \cos(q_1) & 0 & -\sin(q_1) & 0 \\ \sin(q_1) & \cos(q_1) & 0 & 0 & 0 & 1 & 0 & 0 & 0 & \cos(-90) & -\sin(-90) & 0 & \sin(q_1) & 0 & \cos(q_1) & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & d_1 & 0 & \sin(-90) & \cos(-90) & 0 & 0 & -1 & 0 & d_1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \end{matrix}$$

H2:

$$\begin{matrix} \cos(q_2) & -\sin(q_2) & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & \cos(q_2) & 0 & \sin(q_2) & 0 \\ \sin(q_2) & \cos(q_2) & 0 & 0 & 0 & 1 & 0 & 0 & 0 & \cos(90) & -\sin(90) & 0 & \sin(q_2) & 0 & -\cos(q_2) & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & d_2 & 0 & \sin(90) & \cos(90) & 0 & 0 & 1 & 0 & d_2 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \end{matrix}$$

H3:

$$\begin{matrix} \cos(q_3) & -\sin(q_3) & 0 & 0 & 1 & 0 & 0 & 0 & \cos(q_3) & 0 & \sin(q_3) & 0 \\ \sin(q_3) & \cos(q_3) & 0 & 0 & 0 & 1 & 0 & 0 & \sin(q_3) & 0 & -\cos(q_3) & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & d_3 & 0 & 0 & 1 & d_3 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \end{matrix}$$

H4:

$$\begin{matrix} \cos(q_4) & -\sin(q_4) & 0 & 0 & 1 & 0 & 0 & 0 & \cos(q_4) & 0 & -\sin(q_4) & 0 \\ \sin(q_4) & \cos(q_4) & 0 & 0 & 0 & 1 & 0 & 0 & \sin(q_4) & 0 & \cos(q_4) & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & d_3 & 0 & 0 & 1 & d_4 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \end{matrix}$$

H5:

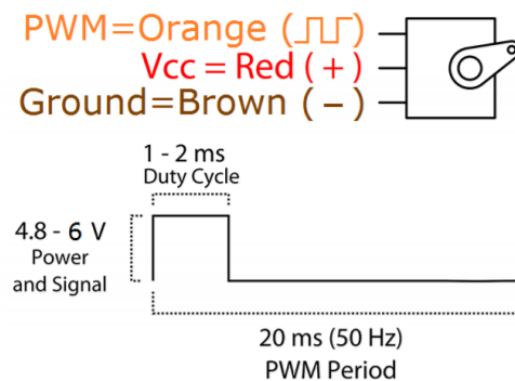
$$\begin{matrix} \cos(q_5) & -\sin(q_5) & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & \cos(q_5) & 0 & -\sin(q_5) & 0 \\ \sin(q_5) & \cos(q_5) & 0 & 0 & 0 & 1 & 0 & 0 & 0 & \cos(-90) & -\sin(-90) & 0 & \sin(q_5) & 0 & \cos(q_5) & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & d_2 & 0 & \sin(-90) & \cos(-90) & 0 & 0 & -1 & 0 & d_5 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \end{matrix}$$

HT:

En proceso

Cálculo de Potencia

Realizamos para nuestro brazo robótico el respectivo cálculo de potencia del miso, para ello se consideró lo siguiente:



Conexiones para control 1

El torque del motor TowerPro MG90S es de 2.5 kgf*cm y sabemos que la velocidad de operación (velocidad angular) es de 60 grados cada 0.1s. Se calcula en base a 4.8 volts que nos suministra el Arduino.



Para el cálculo de potencia:

$$w = \frac{60 \times \frac{2\pi}{360}}{36} = \frac{10\pi}{3} \text{ rad/s}$$

La velocidad angular es de $\frac{10\pi}{3} \text{ rad/s}$ que es igual a 99.99rpm a 4.8 volts de suministro

Obtenemos la potencia en Newton Metro con:

$$M = 1.8 \text{ kg cm} \times \frac{1\text{m}}{100\text{cm}} \times \frac{9.8\text{m}}{\text{s}^2} = 0.176 \text{ Nm}$$

Calculamos la potencia mecánica

$$W = M \times w$$

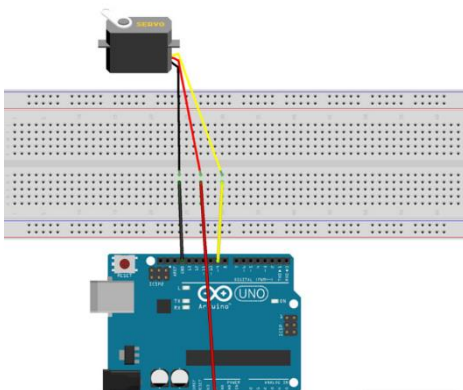
$$0.176 \text{ Nm} \times \frac{\frac{10\pi}{3} \text{ rad}}{\text{s}} = 1.84 \frac{\text{Nm}}{\text{s}}$$

La potencia eléctrica nos da = **1.84 Watts**

Control con Arduino

Los servomotores nos permiten mantener una posición indicada durante un tiempo indicado y más importante aun nos da la posibilidad de controlar a nuestro gusto la velocidad con la que se llega a un ángulo específico de posicionamiento. Es por esto por lo que el control de posición y accionamiento se realizará por medio de servomotores. Cada servomotor estará instalado en una articulación. Para controlar las articulaciones en conjunto se hace uso del microprocesador Arduino que nos facilitará el control de todo el sistema.

El método de control para los servomotores será por medio de señales PWM generadas por el microcontrolador Arduino. El pulso de trabajo varia entre 1ms y 2 ms con un periodo de Ms (50 Hz) lo cual quiere decir que la velocidad máxima de cambio será cada 20 ms (esto depende del tipo de servo).



1.2 Esquema de Conexiones

En el esquema de Conexiones podemos apreciar la conexión entre un Microprocesador Arduino Uno y servo. El pin de control del servo se conecta a un puerto digital del Arduino, dicho puerto deberá contar



con la capacidad de transmitir señales PWM. Los otros dos pines del servo se conectan a tierra y a voltaje de alimentación.

Ejemplo de codificación

```
6 // Incluimos la libreria para poder controlar el servo
7 #include <Servo.h>
8
9 // Declaramos la variable para controlar el servo
10 Servo servoMotor;
11
12 void setup() {
13   // Iniciamos el monitor serie para mostrar el resultado
14   Serial.begin(9600);
15
16   // Iniciamos el servo para que empiece a trabajar con el pin 9
17   servoMotor.attach(9);
18 }
19
20 void loop() {
21   // Desplazamos a la posición 0°
22   servoMotor.write(0);
23   // Esperamos 1 segundo
24   delay(1000);
25
26   // Desplazamos a la posición 90°
27   servoMotor.write(90);
28   // Esperamos 1 segundo
29   delay(1000);
30
31   // Desplazamos a la posición 180°
32   servoMotor.write(180);
33   // Esperamos 1 segundo
34   delay(1000);
35 }
36 }
```

2.1 Ejemplo de codificación

En el código mostrado anteriormente se mueve un servo desde una posición inicial hasta una posición final.

En la línea 7 se incluye la librería para poder habilitar las funciones del servomotor mediante código.

En la línea 10 se le asigna un nombre (servoMotor) al servomotor mediante la instrucción “servo” para poder acceder a el posteriormente y poderlo controlar.

En la línea 17 dentro de la sección Void SetUp configuramos el pin por el cual se conectó físicamente el servo al microprocesador con la instrucción “servomotor.attach(# pin)”

Dentro de la sección de void loop en la línea 23 configuramos la posición inicial en grados de la cuál partirá nuestro servomotor mediante la instrucción “servomotor.write(°ángulo)”. Posteriormente en la línea 25 dejamos que el sistema tenga un respiro para procesar señales y el hardware tenga la posibilidad de estabilizarse. En la línea 28 cambiamos a 90° mediante “servoMotor.write(90)” y posteriormente efectuamos un respiro con un “delay”, esta instrucción de respiro de igual manera dictamina el tiempo en que tardaremos de llegar de una posición otra posición. Este es el proceso de cambiar desde una posición a otra posición lo cual se verá reflejado en el movimiento de articulaciones del brazo.



Costo

Una vez contemplado todo lo anterior procedimos a el cálculo del costo del robot, para ello realizamos la siguiente tabla:

Materiales	Cantidad	Costo	Total
Suporte tipo L	1	80	80
Suporte tipo C	3	100	300
Suporte universal	4	120	480
Servomotor TowerPro	6	125	750
Disco de sujeción	6	40	240
Gripper de Metal para Servo MG995	1	320	320
Total			2170

Referencias

C. Chen, R. Hong and H. Wang, "*Design of a Controlled Robotic Arm*", 2016 3rd International Conference on Green Technology and Sustainable Development (GTSD), Kaohsiung, 2016.

C. Chen, Z. Chen, "A Remote Controlled Robotic Arm That Reads Barcodes and Handles Products", Department of Mechanical Engineering, National Kaohsiung University of Science and Technology, Kaohsiung, 2018.

Vasco F, "*Brazo Robótico de 5 grados de libertad con transmisión de video e interfaz de control inalámbrico*", Universidad de San Buenaventura, Medellín, 2014.

Ollero, A. "*Robótica: Manipuladores y Robots Móviles*", Marcómbi S.A. Barcelona, 2001.

Electronicoscalda. (). Datasheet MG90S. 2020, de TOWER PRO Sitio web:
https://www.electronicoscaldas.com/datasheet/MG90S_Tower-Pro.pdf

Luis del Valle Hernández. (2016). Servomotor con Arduino tutorial de programación paso a paso. 07/12/2020, de Programación fácil Sitio web: <https://programarfácil.com/tutoriales/fragmentos/servomotor-con-arduino>