

Universidad La Salle

Facultad de Ingeniería

Ingeniería Mecatrónica

Fundamento de Robots Manipuladores

Brazo robótico de 4 grados de libertad

Alumnos

Marco Antonio Andrade Tirado

Mario Alberto Cisneros Noriega

Diego Amaury Hernández Sanabria

Luis Antonio Pérez Ponce

Alberto Andrés Sánchez Rodríguez

Profesor

Jesús Enrique Gaytán

18 de diciembre del 2020



Resumen- Se propuso un diseño mecánico de un brazo mecánico, seleccionando elementos mecánicos, materiales, cálculos, ajustes y tolerancias, buscando tener precisión con el trabajo asignado, el brazo cuenta con 4 grados de libertad, resolviendo los cálculos con el método Denavit-Hartenberg para realizar la programación de los cálculos en Matlab.

Abstract- A mechanical design of a mechanical arm was proposed, selecting mechanical elements, materials, calculations, adjustments and tolerances, seeking to have precision with the assigned work, the arm has 4 degrees of freedom, solving the calculations with the Denavit – Hartenberg method to perform program scheduling.

Palabras clave: Robots manipuladores, Denavit-Hartenberg, grados de libertad, comunicación, mecánico.

1. Introducción.

Brazos robóticos, el avance de la tecnología es imparable y trae a menudo diferentes tipos de máquinas e inventos que facilitan los trabajos. Uno de ellos es el brazo robótico que hoy en día se emplea en grandes fábricas. Se emplea para sustituir y ampliar funciones que realizaría un brazo humano. Sus funciones, así como las aplicaciones que puede llegar a tener, se van desarrollando día a día. Permite, de esta manera, que los trabajos mecánicos en las fábricas sean más sencillos para sus trabajadores, además evita que se produzcan accidentes laborales relacionados con la carga de peso, entre otras.

Los tipos de brazos robóticos que puedes encontrar en las industrias y fábricas. Estos facilitan el trabajo de los empleados de las industrias y aumentan el rendimiento de las mismas. Así, permiten realizar tareas complejas de levantamiento o traslados de

pesos que resultan costosos o peligrosos para los humanos.

Robot cartesiano

Este robot se caracteriza por sus ejes, que coinciden con los tres ejes cartesianos. Se emplea en funciones de soldadura, en operaciones de ensamblado, manipulación de objetos e incluso aplicación de impermeables.

Robot esférico o polar

Los ejes de este robot forman un sistema polar de coordenadas. Es decir, en un círculo un eje que va de arriba a abajo y otro de derecha a izquierda (como norte-sur este-oeste). Su origen se encuentra en el primer robot Unimate instalado en una fábrica coches, la General Motors. Este realizaba trabajos de soldadura, fundición y manipulación de máquinas y herramientas.

Robot articulado

El robot articulado tiene, como mínimo, tres articulaciones que giran sobre sí mismas, lo que le permite llevar a cabo tareas más complejas. La mayor rotación permite que pueda realizar desde soldaduras hasta pintar con spray. Se emplea, con frecuencia, en la industria automovilística.

Robot cilíndrico

Sus ejes forman un sistema de coordenadas de círculos concéntricos que le permiten efectuar tareas como la manipulación de máquinas. Pero, además, puede realizar funciones de soldadura de punto. También manejan maquinaria de fundición a presión y operaciones de ensamblaje.

Robot SCARA

Es un robot con dos articulaciones rotatorias paralelas, que permiten que pueda hacer trabajos de pick and place, que significa coger y dejar. Es decir, las principales funciones de

este tipo de brazo son las de recoger objetos y dejarlos en otros lugares. En este caso tienes que fijarte en la fuerza que pueden tener estas máquinas. Pueden recoger metales pesados o estructuras imposibles de mover por una sola persona y moverlas.

En la automovilística, el ensamblaje o ajuste de piezas es un lugar recurrente para emplazar estas máquinas. Sin embargo, también pueden servir para el pick and place. Es decir, para transportar materiales pesados de un lugar a otro o colocarlos dentro de la estructura del automóvil.

Además, los brazos con rotaciones más perfeccionadas pueden realizar trabajos de pintado de los coches una vez terminados.

Pero esta industria no es la única ya que el brazo robótico ha llegado hasta el espacio. En el Transbordador Espacial se encuentra un Sistema de Manipulación Remota que es un conjunto de brazos mecánicos. Estos permiten controlar el sistema por los humanos desde otro punto. Sus funciones van desde la inspección de satélites hasta la carga y descarga de material.

El proyecto consiste en realizar a escala un brazo robótico que cuente con 4 grados de libertad, al final el robot será capaz de moverse en diferentes posiciones. Estará programado y con los cálculos pertinentes para su correcto funcionamiento.

2. Objetivo

Se diseñarán los programas pertinentes para la realización de cálculos de un brazo robótico el cual será simulado en el software Tinkercad. En él se colocarán 5 servomotores conectados al Arduino para ser manipulados de dos formas, una por software y otra por hardware.

De manera análoga, aplicar los conocimientos pertinentes obtenidos en clase de manera

previa. Aplicaremos estos conocimientos tanto teóricos como prácticos para la realización de un robot. De esta manera lograremos manipularlo a nuestra conveniencia y comprobaremos los conocimientos que hemos obtenido hasta ese punto.

3. Estado del Arte

El documento encontrado, titulado REDISEÑO DE BRAZO MECÁNICO PARA PRÁCTICAS DE LABORATORIO DE MECATRÓNICA, César Reyes Rodríguez¹, José de Jesús Robledo Córdova², José Luís Viramontes Reyna², Edgar Froylan Valdés Mata¹ [1] Alumno de Posgrado, [2] Universidad Tecnológica de San Luis Potosí. Pról. Av. de las Américas No.100, Rancho Nuevo Soledad de Graciano Sánchez, S.L.P. México; Apartado Postal 1-16, Código Postal 78430. presenta el rediseño mecánico de un brazo, el cálculo de torque en el brazo para selección del motor adecuado y distribución del centro de masas para elementos ensamblados, el cálculo es seleccionando ajustes usando interferencia, selección de materiales con baja densidad, alta resistencia, y realización de planos como apoyo. El proyecto permite a los alumnos aplicar las estrategias para realización de prácticas y proyectos utilizando los brazos en su totalidad como complemento en su formación.

4. Marco teórico

Antecedentes

El concepto de máquinas automatizadas se remonta a la antigüedad, con mitos de seres mecánicos vivientes. Los autómatas, o máquinas semejantes a personas, ya aparecían en los relojes de las iglesias medievales, y los relojeros del siglo XVIII eran famosos por sus ingeniosas criaturas mecánicas. Algunos de los primeros robots empleaban mecanismos de realimentación

para corregir errores, mecanismos que siguen aplicándose actualmente. Un ejemplo de control por realimentación es un bebedero que emplea un flotador para determinar el nivel del agua. Cuando el agua cae por debajo de un nivel determinado, el flotador baja, abre la válvula y deja entrar más agua en el bebedero. Al subir el agua el flotador también sube, y al llegar a cierta altura se cierra la válvula y se corta el paso del agua. El primer auténtico controlador realimentado fue el regulador de Watt, inventado en 1788 por el ingeniero británico James Watt. Este dispositivo constaba de dos bolas metálicas unidas al eje motor de una máquina de vapor y conectadas con una válvula que regulaba el flujo de vapor. A medida que aumentaba la velocidad de la máquina de vapor, las bolas se alejaban del eje debido a la fuerza centrífuga, con lo que cerraban la válvula. Esto hacía que disminuyera el flujo de vapor a la máquina y por tanto la velocidad. El control por realimentación, el desarrollo de herramientas especializadas y la división del trabajo en tareas más pequeñas que pudieran realizar obreros o máquinas fueron ingredientes esenciales en la automatización de las fábricas en el siglo XVIII. A medida que mejoraba la tecnología se desarrollaron máquinas especializadas para tareas como poner tapones a las botellas o verter caucho líquido en moldes para neumáticos. Sin embargo, ninguna de estas máquinas tenía la versatilidad del brazo humano, y no podían alcanzar objetos alejados y colocarlos en la posición deseada. El desarrollo del brazo artificial multiarticulado, o manipulador, llevó al moderno robot.

Robots Manipuladores

Un robot industrial es un manipulador multifuncional reprogramable, capaz de mover materias, piezas, herramientas, o dispositivos especiales, según trayectorias

variables, programadas para realizar tareas diversas. El campo de la robótica industrial puede definirse como el estudio, diseño y uso de robots para la ejecución de procesos industriales.

Manipulador multifuncional reprogramable con varios grados de libertad, capaz de manipular materias, piezas, herramientas o dispositivos especiales según trayectorias variables programadas para realizar tareas diversas.

la Federación Internacional de Robótica (IFR) distingue entre robot industrial de manipulación y otros robots:

“Por robot industrial de manipulación se entiende una máquina de manipulación automática, reprogramable y multifuncional con tres o más ejes que pueden posicionar y orientar materias, piezas, herramientas o dispositivos especiales para la ejecución de trabajos diversos en las diferentes etapas de la producción industrial, ya sea en una posición fija o en movimiento.”

Denavit-Hartenberg

Se trata de un procedimiento sistemático para describir la estructura cinemática de una cadena articulada constituida por articulaciones con un solo grado de libertad.

Para ello, a cada articulación se le asigna un Sistema de Referencia Local con origen en un punto Q_i y ejes ortonormales $\{X_i, Y_i, Z_i\}$, comenzando con un primer S.R fijo e inmóvil dado por los ejes $\{X_0, Y_0, Z_0\}$, anclado a un punto fijo Q_0 de la Base sobre la que está montada toda la estructura de la cadena.

Este Sistema de Referencia no tiene por qué ser el Universal con origen en $(0,0,0)$ y la Base canónica

Grados de libertad

Los grados de libertad son el número mínimo de velocidades generalizadas independientes necesarias para definir el estado cinemático de un mecanismo o sistema mecánico. El número de grados de libertad coincide con el número de ecuaciones necesarias para describir el movimiento.

Protocolo I2C

I2C es un puerto y protocolo de comunicación serial. Define la trama de datos y las conexiones físicas para transferir bit entre dos dispositivos digitales.

El I2C es un protocolo síncrono. I2C usa solo 2 cables, uno para el reloj (SCL) y otro para el dato (SDA). Esto significa que el maestro y el esclavo envían datos por el mismo cable, el cual es controlado por el maestro, que crea la señal de reloj. I2C no utiliza selección de esclavo, sino direccionamiento.

La principal característica de I²C es que utiliza dos líneas para transmitir la información: una para los datos y otra para la señal de reloj. También es necesaria una tercera línea, pero esta sólo es la referencia (masa). Como suelen comunicarse circuitos en una misma placa que comparten una misma masa esta tercera línea no suele ser necesaria.

Git

GitHub es una forja para alojar proyectos utilizando el sistema de control de versiones Git. Se utiliza principalmente para la creación de código fuente de programas de ordenador.

Git es un software de control de versiones diseñado por Linus Torvalds, pensando en la eficiencia y la confiabilidad y compatibilidad del mantenimiento de versiones de aplicaciones cuando éstas tienen un gran número de archivos de código fuente.

Al principio, Git se pensó como un motor de bajo nivel sobre el cual otros pudieran escribir la interfaz de usuario o front end como Cogito o StGIT. Sin embargo, Git se ha convertido desde entonces en un sistema de control de versiones con funcionalidad plena. Hay algunos proyectos de mucha relevancia que ya usan Git, en particular, el grupo de programación del núcleo Linux.

Comunicación I2C entre Arduino y Matlab

I2C, o Inter-Integrated Circuit, es un protocolo de chip a chip para comunicarse con periféricos de baja velocidad. El paquete de soporte MATLAB[®] para hardware Arduino[®] incluye la biblioteca I2C, que crea una interfaz para comunicarse con los dispositivos I2C. Cada placa Arduino tiene pines específicos para la interfaz I2C. Consulte las especificaciones de su hardware para ubicar los pines correctos.

Puede utilizar dispositivos I2C en muchas aplicaciones, que incluyen:

- Relojes en tiempo real
- Potenciómetros digitales
- Sensores de temperatura
- Brújulas digitales
- Chips de memoria
- Circuitos de radio FM
- Expansores de entrada / salida
- Controladores LCD
- Amplificadores

Los dispositivos Arduino tienen uno o dos buses I2C. Cada bus tiene un I2C Master conectado a dos líneas bidireccionales, una línea de datos en serie (SDA) y un reloj en serie (SCL). Estas dos líneas están conectadas a un par de pines en el hardware. Puede conectar varios dispositivos I2C, como ADC, LCD y sensores, a los pines I2C en el hardware Arduino. Cada dispositivo I2C en un bus I2C debe tener una dirección única. La mayoría de

los dispositivos tienen una dirección predeterminada asignada por el fabricante. Si la dirección no es única, consulte la hoja de datos del dispositivo y vuelva a configurar la dirección. A menudo, puede reconfigurar la dirección usando un par de puentes en el dispositivo. El paquete de soporte de MATLAB para hardware Arduino solo admite direccionamiento de 7 bits.

Arduino

Arduino es una compañía de desarrollo de software y hardware libres, así como una comunidad internacional que diseña y manufactura placas de desarrollo de hardware para construir dispositivos digitales y dispositivos interactivos que puedan detectar y controlar objetos del mundo real. Arduino se enfoca en acercar y facilitar el uso de la electrónica y programación de sistemas embebidos en proyectos multidisciplinarios. Los productos que vende la compañía son distribuidos como Hardware y Software Libre, bajo la Licencia Pública General de GNU (GPL) y la Licencia Pública General Reducida de GNU (LGPL)

Arduino es una plataforma electrónica de código abierto basada en hardware y software fáciles de usar. Las placas Arduino pueden leer entradas (luz en un sensor, un dedo en un botón o un mensaje de Twitter) y convertirlo en una salida, activando un motor, encendiendo un LED, publicando algo en línea. Puede decirle a su tablero qué hacer enviando un conjunto de instrucciones al microcontrolador en el tablero. Para hacerlo, utiliza el lenguaje de programación Arduino (basado en Wiring) y el Software Arduino (IDE), basado en Processing.

Arduino nació en el Ivrea Interaction Design Institute como una herramienta fácil para la creación rápida de prototipos, dirigida a estudiantes sin experiencia en electrónica y

programación. Tan pronto como llegó a una comunidad más amplia, la placa Arduino comenzó a cambiar para adaptarse a las nuevas necesidades y desafíos, diferenciando su oferta desde placas simples de 8 bits hasta productos para aplicaciones de IoT, wearable, impresión 3D y entornos integrados. Todas las placas Arduino son completamente de código abierto, lo que permite a los usuarios construirlas de forma independiente y eventualmente adaptarlas a sus necesidades particulares. El software también es de código abierto y está creciendo gracias a las contribuciones de los usuarios de todo el mundo.

Tinkercad.

Tinkercad es un programa de modelado 3D en línea gratuito que se ejecuta en un navegador web, conocido por su simplicidad y facilidad de uso. Desde que estuvo disponible en 2011, se ha convertido en una plataforma popular para crear modelos para impresión 3D, así como una introducción de nivel de entrada a la geometría sólida constructiva en las escuelas.

Tinkercad no solo sirve para el modelado en 3D en línea, igualmente se ejecuta como un programa de armado, modelado y simulación de circuitos eléctricos. Hecho por Autodesk, Inc.

Tinkercad fue fundada como empresa en 2010 en la Unión Europea por el ex ingeniero de Google Kai Backman y su cofundador Mikko Mononen, con el objetivo de hacer que el modelado 3D, especialmente el diseño de elementos físicos, sea accesible para el público en general y permita que los usuarios publiquen sus diseños bajo una licencia Creative Commons.

Servomotores.

Un servomotor es un tipo especial de motor que permite controlar la posición del eje en un momento dado. Está diseñado para moverse determinada cantidad de grados y luego mantenerse fijo en una posición.

Un servomotor (o servo) es un tipo especial de motor con características especiales de control de posición. Al hablar de un servomotor se hace referencia a un sistema compuesto por componentes electromecánicos y electrónicos.

Está conformado por un motor y un circuito de control. También potencia proporcional para cargas mecánicas. Un servo, por consiguiente, tiene un consumo de energía reducido.

La corriente que requiere depende del tamaño del servo. Normalmente el fabricante indica cuál es la corriente que consume. La corriente depende principalmente del par, y puede exceder un amperio si el servo está enclavado.

En otras palabras, un servomotor es un motor especial al que se ha añadido un sistema de control (tarjeta electrónica), un potenciómetro y un conjunto de engranajes. Con anterioridad los servomotores no permitían que el motor girara 360 grados, solo aproximadamente 180; sin embargo, hoy en día existen servomotores en los que puede ser controlada su posición y velocidad en los 360 grados. Los servomotores son comúnmente usados en modelismo como aviones, barcos, helicópteros y trenes para controlar de manera eficaz los sistemas motores y los de dirección.

Los servomotores hacen uso de la modulación por ancho de pulsos (PWM) para controlar la dirección o posición de los motores de corriente continua. La mayoría trabaja en la frecuencia de los 50 hertz, así las señales PWM tendrán un periodo de veinte

milisegundos. La electrónica dentro del servomotor responderá al ancho de la señal modulada. Si los circuitos dentro del servomotor reciben una señal de entre 1 a 1,4 milisegundos, este se moverá en sentido horario; entre 1,6 a 2 milisegundos moverá el servomotor en sentido antihorario; 1,5 milisegundos representan un estado neutro para los servomotores estándares.

Potenciómetros.

Un potenciómetro es un resistor eléctrico con un valor de resistencia variable y generalmente ajustable manualmente. Los potenciómetros utilizan tres terminales y se suelen utilizar en circuitos de poca corriente, para circuitos de mayor corriente se utilizan los reóstatos. En muchos dispositivos eléctricos los potenciómetros son los que establecen el nivel de salida. Por ejemplo, en un altavoz el potenciómetro ajusta el volumen; en un televisor o un monitor de ordenador se puede utilizar para controlar el brillo.

Los potenciómetros limitan el paso de la corriente eléctrica (Intensidad) provocando una caída de tensión en ellos al igual que en una resistencia, pero en este caso el valor de la corriente y la tensión en el potenciómetro las podemos variar solo con cambiar el valor de su resistencia. En una resistencia fija estos valores serían siempre los mismos

El valor de un potenciómetro viene expresado en ohmios (símbolo Ω) como las resistencias, y el valor del potenciómetro siempre es la resistencia máxima que puede llegar a tener. El mínimo lógicamente es cero. Por ejemplo, un potenciómetro de 10K Ω puede tener una resistencia variable con valores entre 0 Ω y 10.000 Ω .

Servo.h

Permite que las placas Arduino / Genuino controlen una variedad de servomotores.

Esta biblioteca puede controlar una gran cantidad de servos. Hace un uso cuidadoso de los temporizadores: la biblioteca puede controlar 12 servos usando solo 1 temporizador. En el Arduino Due puedes controlar hasta 60 servos.

Esta biblioteca permite que una placa Arduino controle servomotores RC (hobby). Los servos tienen engranajes integrados y un eje que se puede controlar con precisión. Los servos estándar permiten colocar el eje en varios ángulos, generalmente entre 0 y 180 grados. Los servos de rotación continua permiten ajustar la rotación del eje a varias velocidades.

Para usar esta biblioteca:

```
#include <Servo.h>
```

Los servomotores tienen tres cables: alimentación, tierra y señal. El cable de alimentación suele ser rojo y debe conectarse al pin de 5 V de la placa Arduino. El cable de tierra suele ser negro o marrón y debe conectarse a un pin de tierra en la placa Arduino. El pin de señal suele ser amarillo, naranja o blanco y debe conectarse a un pin digital en la placa Arduino.

5. Planteamiento del problema

Realizar un diseño factible de la estructura de un brazo mecánico conociendo los elementos disponibles en el mercado para visualizar y lograr un modelo más acertado de la estructura que realizaremos junto con una idea clara de cuál es el funcionamiento del sistema. Realizar los cálculos pertinentes con el método Denavit–Hartenberg, para poder realizar la programación y armado de un brazo robótico.

En cuando a la construcción del brazo robótico de acuerdo al diseño previo

utilizando los materiales, componentes y herramientas para montar la estructura, en este caso se realizo el proyecto solo con programas de simulación, en dado caso de realizar el proyecto en físico se enlistarán los materiales para así adaptar todos los elementos y actuadores en el sistema.

Programación del sistema para que cumpla con un funcionamiento adecuado es obtener una idea clara de la estructura de la programación y como vamos a hacer la conexión I2C y la conexión entre Matlab y Arduino.

6. Desarrollo.

Para comenzar el proyecto vamos a proponer el modelo del robot de 4 grados de libertad.

Ya teniendo una base del modelo se procede al modelado del robot para tener una mejor idea del movimiento y las acciones que va a realizar. Este modelo se realizó en Tinkercad Diseño en 3D.

Desarrollo del programa de Denavit-Hartenberg

```
clear all
close all
clc
prompt = {'Ingresa el numero de
brazos'};%solicitamos una entrada
dlg_title = 'Entrada';%nombre del
título
num_lines = 1;%definimos el numero
de líneas de la entrada
def = {'1'};%definimos el número
por default que aparece en la
entrada
answer =
inputdlg(prompt,dlg_title,num_lines
,def);%guardamos los valores
obtenidos en matriz
num = str2num(answer{:});%guardamos
y cambiamos los valores en vectores
```



```

F = sym('A', [num 4]);%creamos una
matriz simbolica con los datos de
num guardados
B=eye(4);%creamos una matriz con
diagonales con 1
C = sym('C', [4 4]);%creamos una
matriz c de 4x4 con c de prefijo
clc%limpiamos la ventana de
comandos
for i=1:num%iniciamos un ciclo
hasta el numero especificado
    prompt = {'Ingresa a:', 'Ingresa
alfa:', 'Ingresa d:', 'Ingresa
theta:'};%desplegamos para pedir
informacion de cada brazo
    dlg_title =
sprintf('brazo%d',i);%le ponemos
nombre para cada brazo
    num_lines = 1;%definimos el
numero de lineas
    def1 =
{sprintf('a%d',i),sprintf('alfa%d',
i),sprintf('d%d',i),sprintf('t%d',i
)};%datos de cada brazo
    answer1 =
inputdlg(prompt,dlg_title,num_lines
,def1);%los almacenamos en answer 1

F(i,1)=answer1(1,1);%almacenamos
los valores en una matriz F
    F(i,2)=answer1(2,1);
    F(i,3)=answer1(3,1);
    F(i,4)=answer1(4,1);
    C=simplify([cos(F(i,4)) -
sin(F(i,4))*cos(F(i,2)) -
sin(F(i,4))*sin(F(i,2))
F(i,1)*cos(F(i,4));
            sin(F(i,4))
cos(F(i,4))*cos(F(i,2)) -
cos(F(i,4))*sin(F(i,2))
F(i,1)*sin(F(i,4));
            0 sin(F(i,2))
cos(F(i,2)) F(i,3);
            0 0 0 1]);%se hace el
calculo de la ultima matriz
    eval(sprintf('A%d =
C;',i));%evaluamos y pasamos el
valor a A%d
    B=B*C;
    eval(sprintf('A%d',i))

```

```

end
sprintf('Matriz del brazo 0 al
brazo %d is:',i)
pretty(simplify(B))%imprimimos la
matriz simplificada
'Matriz de rotacion' ,
R=B(1:3,1:3);%separamos los valores
de la matriz
pretty(R)
d=B(1:3,4);
'Matriz de traslacion' ,
pretty(d)%imprimimos solo la ultima
columna resultante

```

Desarrollo programa de comunicación entre Matlab y Arduino.

```

clear port
clear all port
%para saber donde esta conectado el
arduino
global a;
a=arduino('COM9','leonardo');
%Movimiento de cada motor
%Los angulos van de 0 a 1
% 0 = 0° ; 1 = 180°
% Conversión de angulo ->
(0.555*grados)/100
s1=servo(a,3);
writePosition(s1,0);
s2=servo(a,4);
writePosition(s2,0.5);
s3=servo(a,5);
writePosition(s2,1);
s4=servo(a,6);
writePosition(s2,0.5);

```

7. Modelo

Materiales y sus características

Para los materiales que se utilizaran partiremos de dos ramas, un que sean materiales fáciles de conseguir para en cierto punto hacer el modelo del brazo en casa para no ser expuestos a la contingencia actual y, por otro lado, materiales que se pueden conseguir en el taller de la universidad.

Se ha optado en usar la madera (madera balsa para maquetas) a la hora de construir el robot ya que se trata de un material asequible, de fácil uso y, además, económico. En cuanto a los motores se han elegido servos, debido a los mismos motivos que la madera, así como su fácil programación.

La madera de balsa es una madera poco nerviosa con una densidad de 140-160 kg/m³ y muy blanda. Sus propiedades mecánicas son las siguientes:

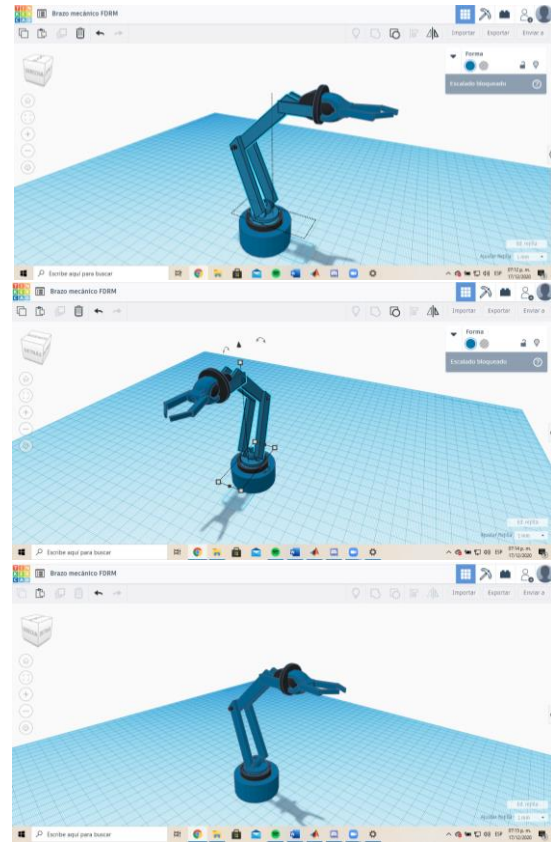
- Flexión estática: 15-23 N/mm²
- Módulo de elasticidad: 2160-3200 N/mm²
- Compresión axial: 5-15 N/mm²
- Compresión perpendicular: 2,6 N/mm²

Para un modelo hecho en un taller con las debidas normas y poder optimizar y facilitar el proceso reconstructivo del brazo mecánico, seleccionamos como material el aluminio, elegimos este material ya que posee una adecuada resistencia mecánica para las condiciones que son requeridas en el trabajo, tiene una sencilla maquinabilidad, una densidad menor que el acero que permite reducir el peso.

- Características físicas:
 - Es un metal ligero, cuya densidad o peso específico es de 2700 kg/m³.
 - Tiene un punto de fusión bajo: 660 °C (993 K)
 - El peso atómico del aluminio es de 26,9815.
 - Buen conductor de calor y electricidad
- Características mecánicas:
 - Fácil mecanizado
 - Muy maleable
 - Bastante dúctil, blando y soldable

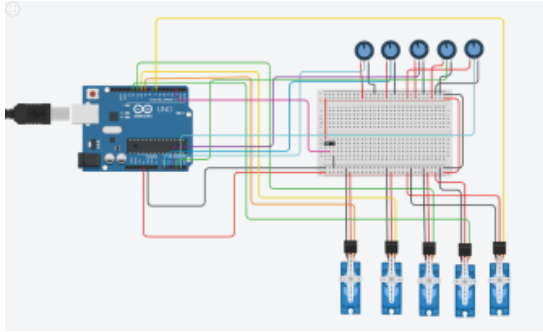
La resistencia de cizallamiento que permite seleccionar el material de acuerdo al resultado obtenido del cálculo de esfuerzo de cizallamiento o cortante.

Modelo del brazo



8. Resultados

Como podemos observar la siguiente imagen se logró la simulación de un programa que es capaz de leer valores desde el monitor serial seleccionando el modo Matlab con el botón. También implementamos un teachpendant muy arcaico con ayuda de los potenciómetros. Para la programación fue en lenguaje Arduino y en el modo Matlab leemos los valores del monitor serial en una matriz y después los enviamos a los servomotores para el segundo modo leemos los valores analógicos y se hace un mapeo para después enviar ese valor a los servomotores.



Los cálculos obtenidos para la cinemática de nuestro robot fueron hechos con ayuda del programa de Matlab.

Los resultados obtenidos de la cinemática directa son:

`ans = 'Matriz de traslacion'`

$$x = a_1 \cos(t_1) + a_3 \cos(t_3) - a_3 \sin(t_3) + a_2 \cos(t_1) \cos(t_2) - a_2 \sin(t_1) \sin(t_2) + a_4 \cos(t_4) - a_4 \sin(t_4)$$

$$y = a_1 \sin(t_1) + a_3 \cos(t_3) + a_3 \sin(t_3) + a_2 \cos(t_1) \sin(t_2) + a_2 \cos(t_2) \sin(t_1) + a_4 \cos(t_4) + a_4 \sin(t_4).$$

Conclusiones.

Podemos concluir que fue un proyecto complejo, pero con ayuda de nuestros conocimientos y del profesor, pudimos completar y terminar la simulación del proyecto. Por un lado, tenemos los programas de Matlab. En el primer programa se realizan los cálculos pertinentes para las matrices denavit-hartenberg. En el segundo programa tenemos los algoritmos necesarios para poder enviar los datos obtenidos de los ángulos al Arduino. Para el último programa tenemos la simulación de la obtención de estos ángulos que se reciben y mueven a los servomotores.

9. Referencias.

[1] Craig, J. (2005). *Introduction to robotics: mechanics and control*. USA: Pearson Education.

[2] Genta, G. (2012). *Introduction to the mechanics of space robots*. Netherlands: Springer.

[3] Glaser, A. (2009). *Industrial robotics: how to implement the roght system in your plant*. USA: Industrial

[4] Subik K. (2010). *Introducción a la robótica*. México: McGraw Hill.

[5] Reza, J. (2010). *Theory of applied robotics: kinematics, dynamics, and control*. USA: Springer.

[6] Arámburo, J. y A. Ramírez (2008). *Advances in robotics, automation and control*. Austria: Intech.

[7] Arduino. (2018). *What is Arduino?* 19 de diciembre 2020, de Arduino Sitio web: <https://www.arduino.cc/en/Guide/Introduction>

[8] Tinkercad. (2018). *TINKERCAD BLOG*. 19 de diciembre de 2020, de Tinkercad Sitio web: <https://blog.tinkercad.com/>

[9] Antony García González. (2016). *¿Qué es y cómo funciona un servomotor?* 19 de diciembre 2020, de panamahitek Sitio web: <http://panamahitek.com/que-es-y-como-funciona-un-servomotor/>

[10] Arduino. (2016). *Servo*. 19 de diciembre del 2020, de Arduino Sitio web: <https://www.arduino.cc/reference/en/libraries/servo/>

[11] revesconsult. (2018). *Aluminio*. 19 de diciembre 2020, de revesconsult Sitio web: https://www.revesconsult.com/descargas/propiedades_aluminio.pdf

[12] Freddy M. Alonzo, Miguel E. Bravo. (2014). Diseño, construcción y control de un brazo robótico. 19 de diciembre de 2020, de UNIVERSIDAD

SAN FRANCISCO DE QUITO Sitio web:
<http://repositorio.usfq.edu.ec/bitstream/23000/3840/1/112562.pdf>

URL GITHUB:

<https://github.com/LuisAPerez224/ProyectoRobots>