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 Profesor*

*Alberto Andrés Sánchez Rodríguez 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 del programa.**

***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.***

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.

1. ***Objetivo***

El alumno o la alumna diseñara y fabricara un brazo robótico de 4 grados de libertad con su efecto final. Este deberá incluir documento de apoyo técnico donde se especificará las características de diseño mecánico, modelado cinemáticos y dinámico del robot.

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.

1. ***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.

1. ***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 y ejes ortonormales { , , }, comenzando con un primer S.R fijo e inmóvil dado por los ejes { , , }, anclado a un punto fijo 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.

**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.

1. ***Planteamiento del problema***

Realizar los cálculos pertinentes con el método Denavit–Hartenberg, para poder realizar la programación y armado de un brazo robótico.

1. ***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');

%Moviemiento 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);

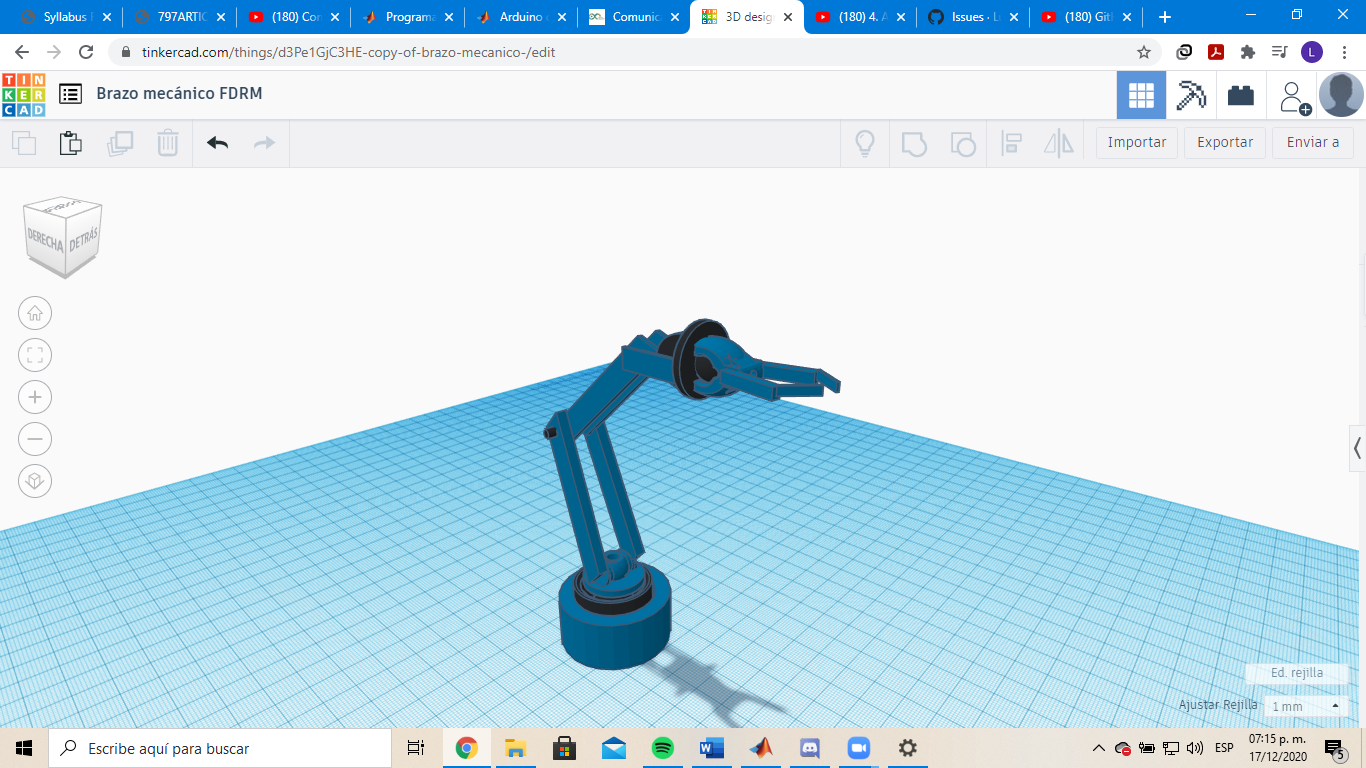
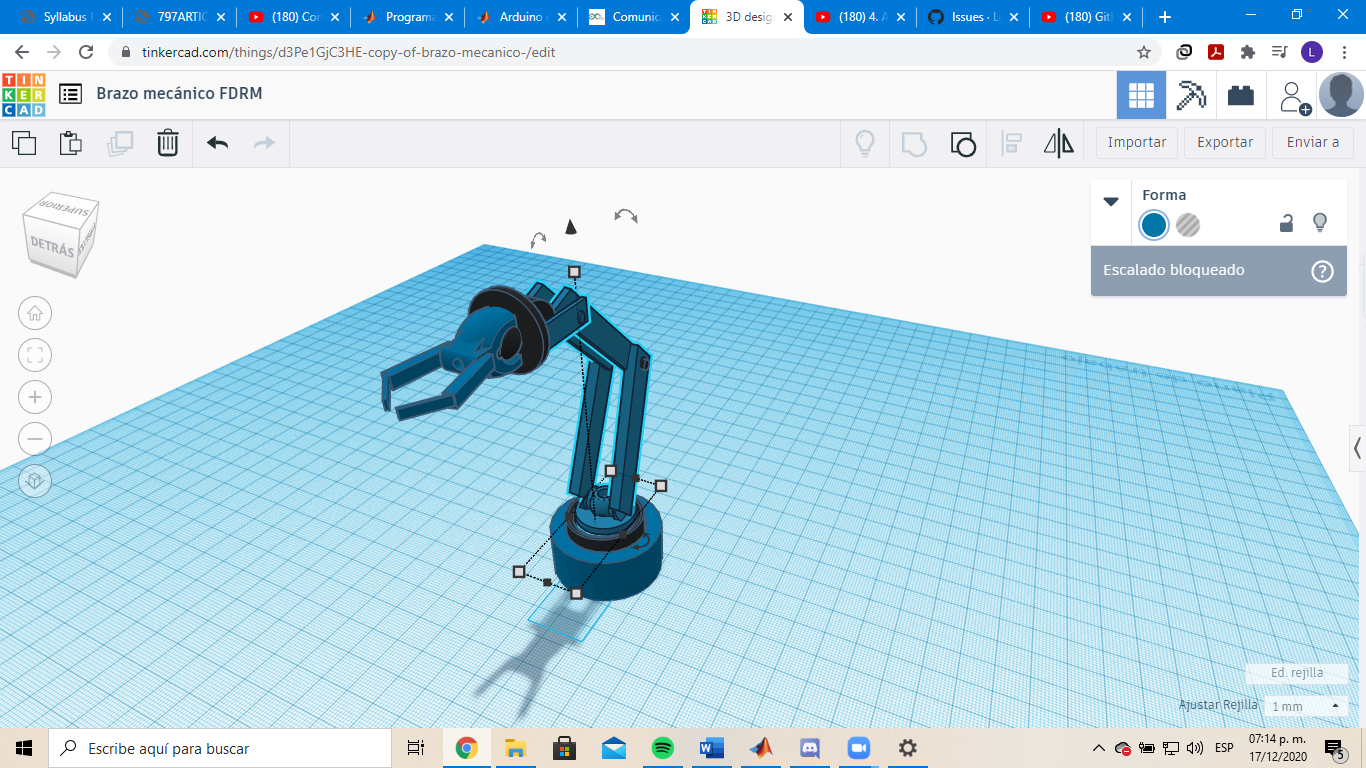
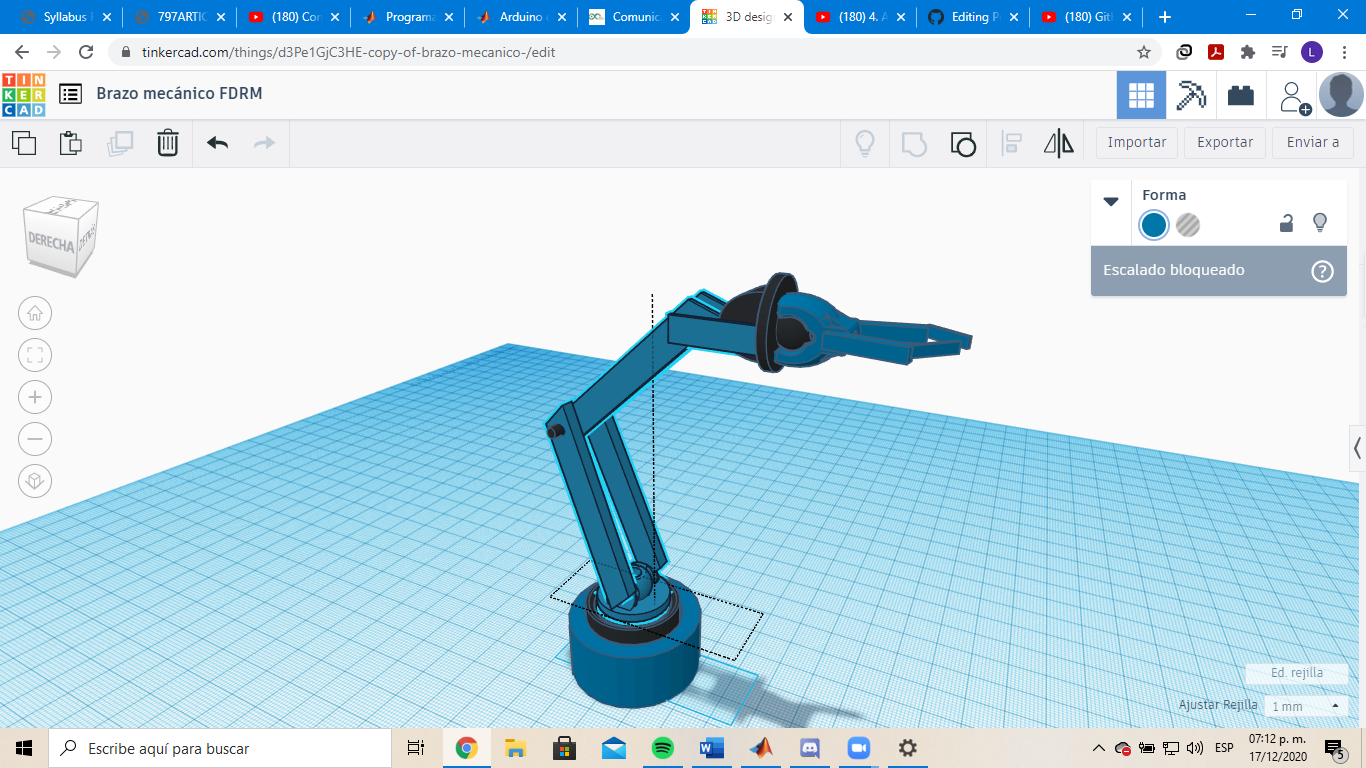
writePosition(s2,1);

s4=servo(a,6);

writePosition(s2,0.5);

1. ***Modelo***

***Modelo del brazo***



1. ***Resultados***
2. ***Conclusiones.***
3. ***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.*