

Universidad Nacional de Trujillo
Escuela de Ingeniería Mecatrónica



ANALISIS Y CONTROL DE UN BRAZO ROBOT 5GDL

INTEGRANTES:

ALVA ALCANTARA JOSMELL

PELAES BAILON ROGER

RUIZ JOAQUIN JORDY

BARRETO VASQUEZ VICTOR

CARRANZA CRUZADO DIDIER

SOBRADOS JHOEL

DOCENTE:

DR. ING. SHLOMI CALDERON.

Trujillo, Perú
2015

I.- INTRODUCCION:

La aparición de la robótica en la industria ha supuesto en los últimos 40 años una importante revolución industrial, que sin embargo no ha tenido el crecimiento exponencial que los pioneros de la robótica habían supuesto. Una de las razones para este no tan rápido crecimiento ha sido la complejidad de los cálculos dinámicos de los mecanismos robotizados. Las elegantes formulaciones matemáticas realizadas por los padres de la dinámica requieren un gran número de cálculos que obligaba a realizar un gran número de simplificaciones.

Históricamente, el análisis, simulación y diseño de robots ha supuesto una gran inversión de tiempo y capital, reservado para centros de investigación y grandes fabricantes.

Sin embargo, la revolución del PC, está permitiendo la implantación de herramientas sencillas de análisis y simulación de robots. Los programas de propósito general para el modelado y la simulación de los sistemas mecánicos de múltiples cuerpos han llamado poderosamente la atención durante esta última década.

Un programa CAD para la dinámica de los sistemas multicuerpos minimiza la posibilidad del error humano en el manejo de modelos complejos, pero en la mayoría de ocasiones, paquetes de software como IMP [Sheth-1972], ADAMS [Orlandea-1977], DISCOS [Bodley-1978], DADS [Nikravesh-1985], MEDYNA [Jaschinsky-1986] ofrecen unas prestaciones mayores de las requeridas por el usuario.

Además, utilizando paquetes de análisis matemático general como MatLab se pueden implementar los algoritmos clásicos de la dinámica computacional, y obtener una serie de herramientas que permitan al usuario del PC realizar un diseño y análisis de su propio mecanismo.

En este sentido, este libro quiere presentar una serie de herramientas sencillas de modelado computacional y simulación de robots, enfocadas a ser utilizadas en el diseño paramétrico de robots y mecanismos tanto plantares como espaciales en computadoras personales. En contraste con los libros clásicos de robótica, en los cuales la teoría ocupa un espacio predominante y es posible encontrar todas las bases de las herramientas que aquí se utilizan, este libro pretende tener un enfoque eminentemente práctico, aportando varias referencias para aquellos lectores que quieran ampliar las bases teóricas.

1.1.- Clasificación de robots. Morfología:

La palabra robot proviene del eslavo *robota* y fue utilizada por primera vez por el Escritor checo Karel Capek en 1917 para referirse en su obra *Rossum's Universal Robot R.U.R.* a máquinas con forma humanoide, pero fue el escritor americano de origen ruso Isaac Asimov el que dotó de popularidad a esta palabra. En octubre de 1945 publicó en la revista *Galaxy Science Fiction* una historia en la que se enuncian por primera vez sus tres leyes de la robótica;

1. Un robot no puede perjudicar a un ser humano, ni con su inacción permitir que un ser humano sufra daño.
2. Un robot ha de obedecer las órdenes recibidas de un ser humano, excepto si tales órdenes entran en conflicto con la primera ley.
3. Un robot debe proteger su propia existencia mientras tal protección no entre en conflicto con la primera o segunda ley. Sin pretender ser exhaustivos con la historia y el desarrollo de la robótica se puede citar que la primera patente de un dispositivo robótico fue solicitada en marzo de 1954 por el inventor británico C.W. Kenward, si bien la primera patente de un *dispositivo de transferencia de artículos programada* que se asemeja más al concepto de robot industrial que hoy conocemos fue presentada por el ingeniero americano George D. Devol en 1961. En ella se dice:

"The present invention makes available for the first time a more or less general purpose machine that has universal application to a vast diversity of applications where cyclic control is desired."

Se pueden distinguir cuatro fechas puntuales en el desarrollo de la robótica industrial:

- En 1948 R.C. Goertz diseñan en el Argonne National Laboratory el primer sistema telemanipulador para manejar materiales radioactivos.
- Joseph F. Engelberger y G.C. Devol fundan en 1958 la primera empresa dedicada a la fabricación de robots industriales; Unimation, e instalan su primera máquina. En 1967 instalaron un conjunto de robots en la factoría de General Motors y tres años después se inicia la implantación de los robots en Europa, especialmente en el sector automovilístico.
- En 1970 la Universidad de Stanford y el MIT empiezan un proyecto destinado a controlar un robot mediante computadora.
- A partir de 1975, la revolución de la microelectrónica y la aparición del microprocesador relanza el sector de la robótica.

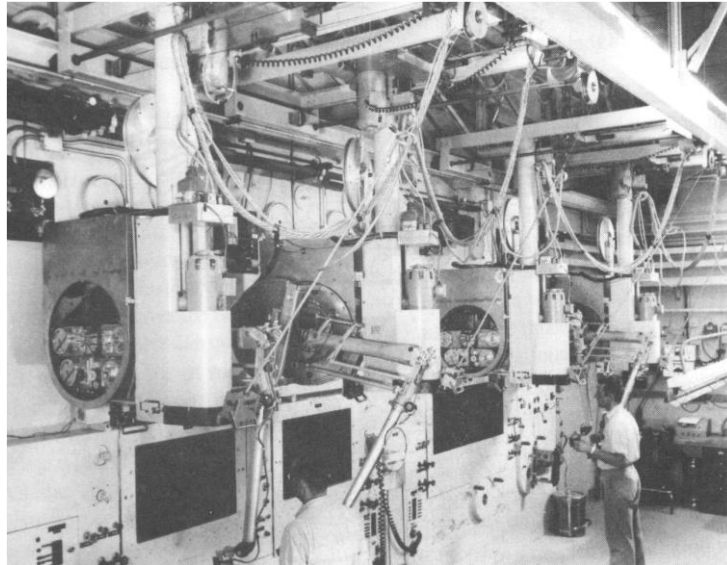


Figura 1.1 Sistema tele operado robótico

A la hora de definir el concepto de robot de distinguir entre el **robot industrial** o robot de producción y los robots especiales o también denominados **robots de servicio**. La definición más comúnmente aceptada de robot industrial es la de la *Asociación de Industrial Robóticas (RIA)*, según la cual:

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.

Mientras que los robots de servicio se definen como:

Dispositivos electromecánicos, móviles o estacionarios, dotados normalmente de uno o varios brazos mecánicos independientes, controlados por un programa de ordenador y que realizan tareas no industriales de servicio para el estudio de la morfología de un robot se debe tener en cuenta que un robot está formado por los siguientes elementos; estructura mecánica, transmisiones, actuadores, sistema sensorial, sistema de control y elementos terminales.

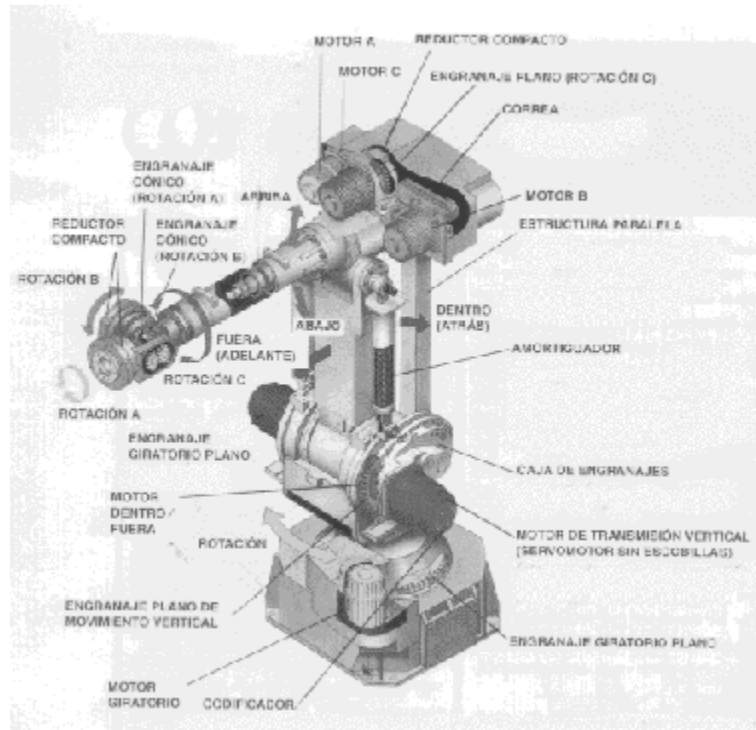


Figura 1.2 Partes de Robots Manipulador

1.2 Estructura mecánica:

Un robot está formado por una serie de elementos o eslabones unidos mediante Articulaciones que permiten un movimiento relativo entre cada dos eslabones consecutivos. La constitución física de la mayor parte de los robots industriales está inspirada en la anatomía del brazo y la mano humana.

Se definen como **grados de libertad** como los posibles movimientos básicos (giros y desplazamientos) independientes. La figura 1.4 muestra diferentes tipos de articulaciones y los grados de libertad que aportan cada una de ellas.

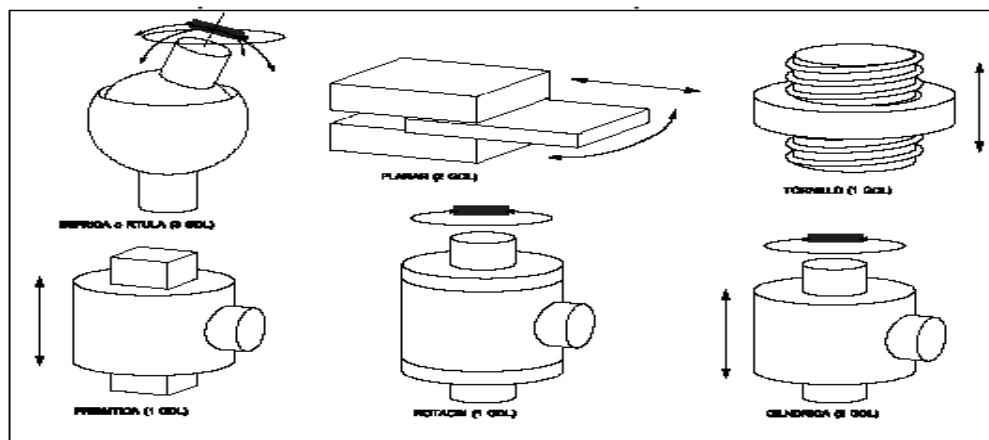


Figura 1.3 Grados de Libertad

Las diferentes combinaciones de estas articulaciones dan lugar a distintas configuraciones o arquitecturas mecánicas, cada una de ellas con características que la hacen más o menos recomendable para la realización de una determinante tarea. Las siguientes figuras muestran las configuraciones más típicas utilizadas en robots industriales, y en ellas se distinguen los eslabones y los grados de libertad de cada configuración.

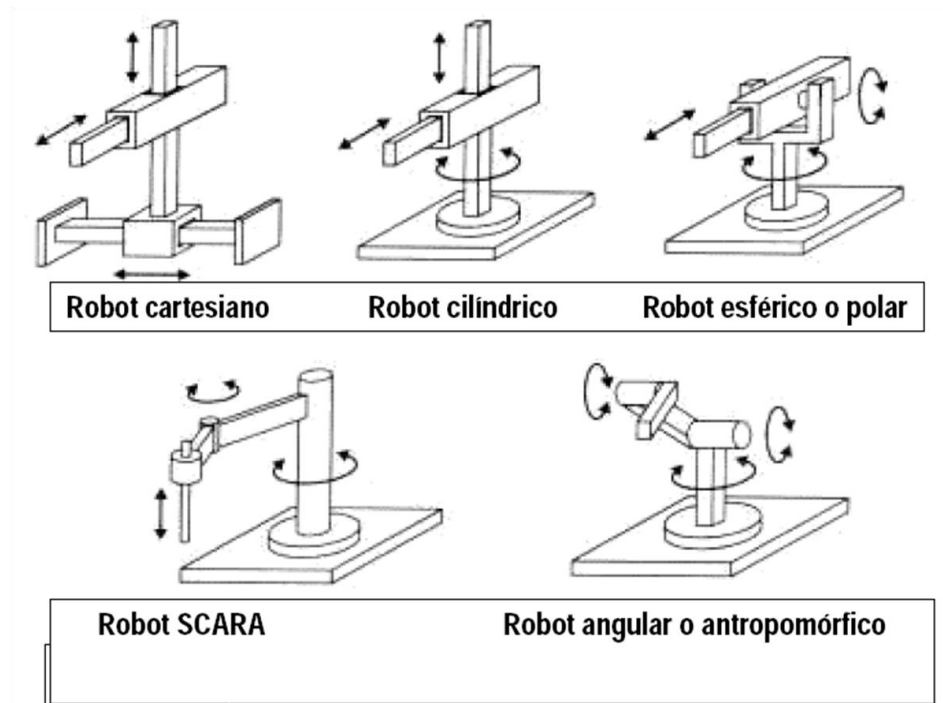


Figura 1.4 tipos de Robots Manipuladores

Es también muy común usar en robótica terminología inspirada en la anatomía del brazo humano, así en la mayoría de arquitecturas mecánicas de robots industriales, las tres últimas articulaciones, que normalmente se utilizan para proporcionar orientación al elemento terminal (se estudiará esto más profundamente en el capítulo siguiente), forman lo que se conoce como **muñeca** del robot.

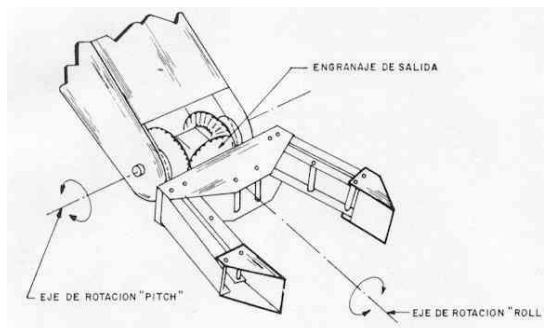


Figura 1.5 Muñeca de Robot Manipulador

Otro concepto importante que debe tenerse en cuenta en la descripción de un robot son las dimensiones del robot y el **espacio de trabajo** que define el movimiento de las diferentes articulaciones.

El espacio de trabajo de un robot se subdivide en áreas diferenciadas entre sí por la accesibilidad del elemento terminal en cada una de ellas.

Por ejemplo, es diferente la zona en la que el elemento terminal solo puede orientarse horizontalmente o aquella en la que también puede orientarse verticalmente. La figura siguiente muestra un robot de la casa STAÜBLI y su espacio de trabajo.

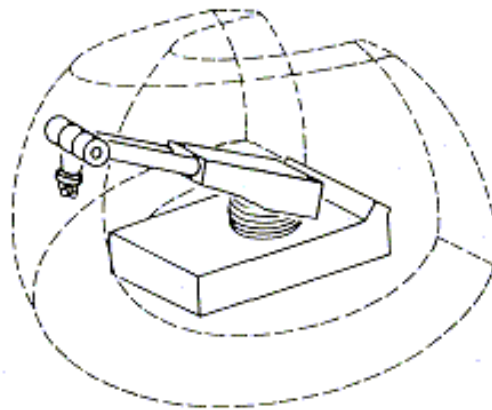


Figura 1.6 Espacio de trabajo de Robot manipulador.

La **capacidad de carga** que puede transportar la garra del robot, la precisión en la **repetibilidad** y la **velocidad** de las articulaciones son también características muy importantes del robot que deben ser citadas en la descripción de un robot.

Las transmisiones son los elementos encargados de transmitir el movimiento desde los actuadores hasta las articulaciones. Los reductores se incluyen con las transmisiones y son los encargados de adaptar el par y la velocidad de la salida del actuador a los valores adecuados para el movimiento de los elementos del robot.

Normalmente los actuadores se intentan situar lo más cerca de la base del robot, con el fin de reducir al máximo el peso estático y la inercia de los eslabones finales, que deben moverse con aceleraciones altas. Para ello es necesario utilizar sistemas de transmisión que trasladen el movimiento hasta las articulaciones. La siguiente tabla, extraída de [1] resume los sistemas de transmisión para robots. El lector que quiera profundizar en el tema puede utilizar las referencias [1],[2] y [3] para obtener más información

Tabla 1. Sistemas de transmisión de Robots

Entrada-Salida	Denominación	Ventajas	Inconvenientes
Circular-Circular	Engranaje	Pares altos	Holguras
	Correa dentada	Distancia grande	---
	Cadena	Distancia grande	ruido
	Paralelogramo	---	giro limitado
	Cable	---	deformabilidad
Circular-lineal	Tornillo sin fin	Poca holgura	Rozamiento
	Cremallera	Holgura media	Rozamiento
Lineal- Circular	Paral. Articulado	---	Control difícil
	Cremallera	Holgura media	Rozamiento

Los reductores utilizados en robótica pueden tener diferentes morfologías y estar basados en distintas tecnologías, y en muchas ocasiones vienen incorporados con el actuador.

1.3 Actuadores:

Para el estudio de los actuadores y sus modelos matemáticos se ha dedicado la práctica cuatro de este libro. Sin embargo el lector debe conocer que los elementos motrices que generan el movimiento de las articulaciones pueden ser, según la energía que consuman, de tipo hidráulico, neumático, eléctrico o basado en las propiedades de las aleaciones con memoria de forma (SMA).

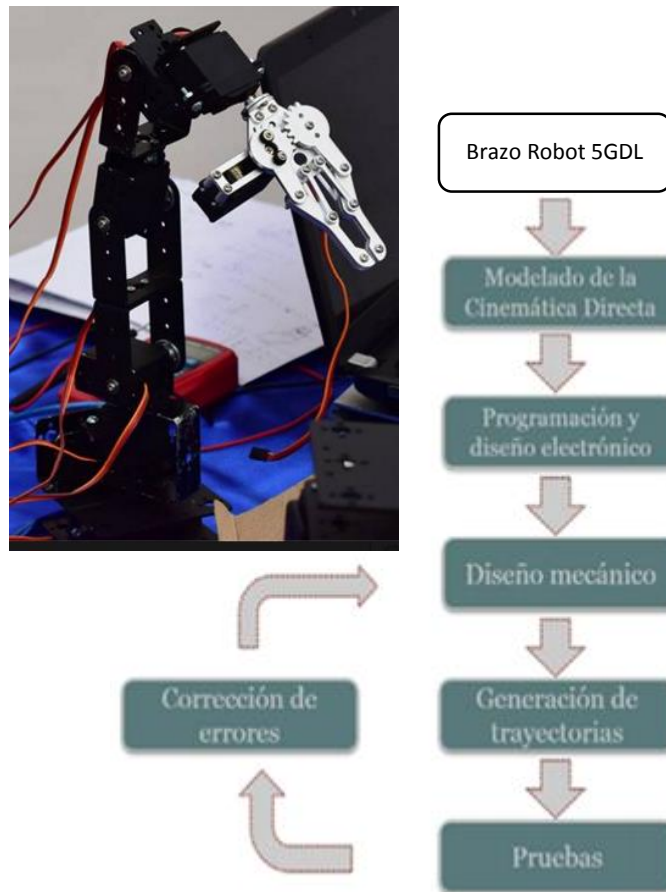


Figura 1.7 Actuadores Roboticos

II .- ESTRATEGIAS:

El objetivo es diseñar y desarrollar un prototipo de un robot manipulador experimental de arquitectura , que permita la libertad de modificar tanto el software como el hardware del brazo robot , dando así a conocer una propuesta del diseño mecánico del robot, utilizando componentes comerciales, de bajo costo y de fácil adquisición; que sea capaz de moverse a las posiciones indicadas para un posterior análisis que nos permita construir una estrategia de control basada en un modelo mecánico (cinemática directa, cinemática inversa, dinámica, control cinemático (planificación de trayectorias) y control dinámico)

1.- PROPUESTA UTILIZADA:



Esquema n°1

2.- ESTRUCTURA MECANICA:

2.1 TIPO DE MATERIAL:

En este capítulo daremos a conocer toda la parte mecánica que se ha utilizado en el diseño del bípedo de 8 grados de libertad con servomotores. Cada pierna del robot tiene 4 grados de libertad. Las piernas pueden realizar giros y movimientos similares a los del humano, ideal para algoritmos de caminado

➤ COMPARACION DE MATERIALES RESPECTO A SUS PROPIEDADES FISICAS

Material	Maleabilidad	Resistencia	Peso
Madera	Baja	Media	Medio
Acrílico	Media	Media	Baja
Aluminio	Alta	Alta	Medio
Plástico	Alta	Medio	Bajo

Tabla1 Tipos de Materiales

ALUMINIO 6061-T6: Aleación dúctil y ligera, con gran resistencia y excelentes características de acabado, el aluminio 6061-T6 es ideal para la elaboración de piezas maquinadas con calidad de excelencia y para trabajos que requieran buen acabado superficial.

Posee excelente resistencia a la corrosión y acabado además de facilidad de soldadura y una resistencia parecida a la del acero. Esta es una aleación de propósito general muy popular con buena facilidad de maquinado a pesar de su tratamiento de envejecimiento artificial (T6).

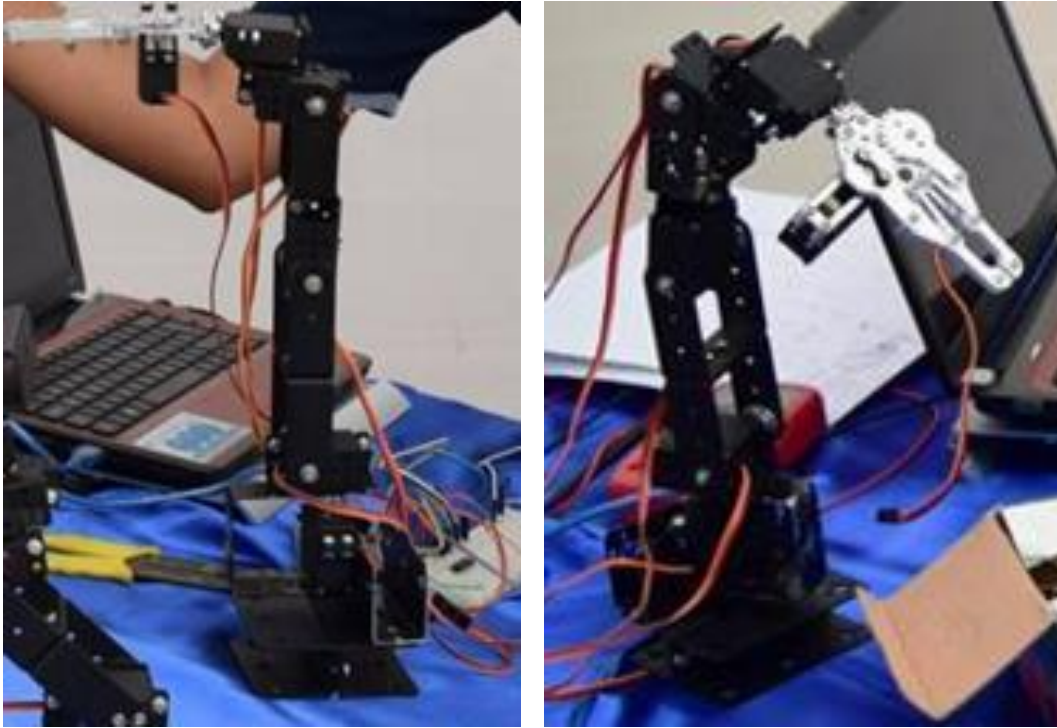


Fig.2.1 Brazo Robot 5 GDL.

2.2 .-TIPOS DE SERVOMOTORES:

Los servomotores, sin importar su tamaño, su torque y su velocidad, se controlan de la misma manera, utilizando la técnica de control PWM (modulación por ancho de pulso). Este sistema consiste en generar una onda cuadrada, en la cual se varía el tiempo en que el pulso se encuentra en nivel alto. Al variar este ancho de pulso se consigue variar el ángulo de giro del servomotor.

Cada servomotor tiene rangos de operación, el mínimo y el máximo ancho de pulso que el servomotor entiende. Los valores más generales de operación varían de 1ms a 2ms de ancho de pulso, los cuales permiten trabajar al mismo entre 0° y 180° respectivamente.

✓ MG996R:

- **Dimensiones** 40.7x19.7x42.9mm
- **Peso** 55g
- **Torque** 9.4Kg (4.8V) ; 11Kg (6.0V)
- **Velocidad** 0.17s/60° (4,8V) ; 0.14s/60° (6.0V)
- **Voltaje de trabajo** 4.8-6.6V
- **Piñonería** Metálica
- **Longitud de cable** 310mm
- **Tipo conector** JR / Universal

- **Rodamiento:** Si



Fig.2.3 servomotor MG996R

2.3 TIPO DE MICRONTROLADOR

➤ ARDUINO MEGA 2560

El Arduino Mega 2560 es una placa con un micro controlador Atmega2560. Cuenta con 54 pines entradas/salidas digitales (de los cuales 14 se pueden utilizar como salidas PWM) y 16 entradas analógicas, 4 UARTs (puertos de hardware de serie), un oscilador de cristal de 16 MHz, un puerto USB de conexión, un conector de alimentación, una cabecera de ICSP, y un botón de reinicio. Contiene todo lo necesario para apoyar al microprocesador, sólo tiene que conectarlo a un ordenador con un cable USB, o con un alimentador adaptador AC-DC o una batería para comenzar. El Arduino Mega es compatible con la mayoría de los escudos diseñados para el Arduino Duemilanove o Diecimila.

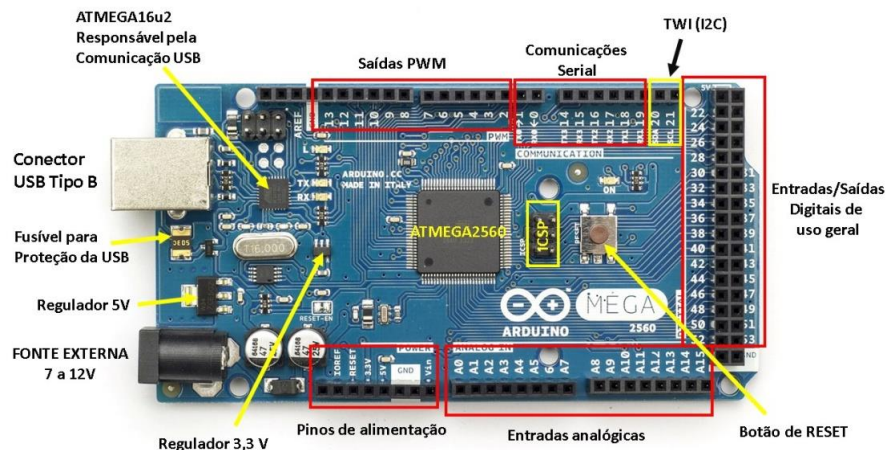


Fig.2.4 Arduino Mega2560

2.4 HERRAMIENTAS VIRTUALES Y SOFTWARES:

2.4.1 SOLIDWORK 2013:

Solidworks Corp., una filial de Dassault Systèmes, S.A. (Suresnes, Francia), para el sistema operativo Microsoft Windows. Su primera versión fue lanzada al mercado en 1995 con el propósito de hacer la tecnología CAD más accesible.

El programa permite modelar piezas y conjuntos y extraer de ellos tanto planos técnicos como otro tipo de información necesaria para la producción.

Es un programa que funciona con base en las nuevas técnicas de modelado con sistemas CAD. El proceso consiste en trasvasar la idea mental del diseñador al sistema CAD, "construyendo virtualmente" la pieza o conjunto. Posteriormente todas las extracciones (planos y ficheros de intercambio) se realizan de manera bastante automatizada.

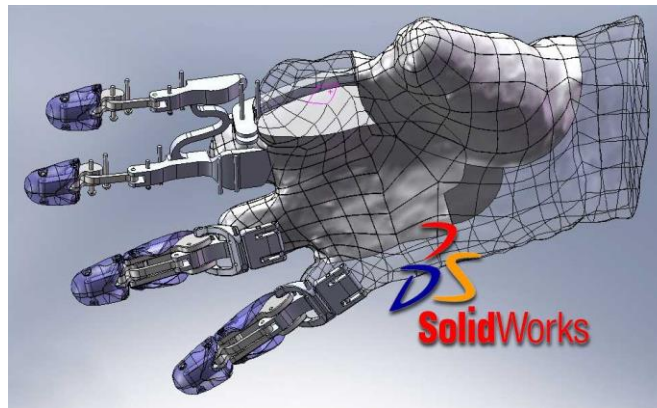


Fig 2.7 SolidWork CAD

2.4.2 ARDUINO IDE:

Arduino también incluye un entorno interactivo de desarrollo (IDE) que permite programar fácilmente la tarjeta. El IDE (Integrated Development Environment) se basa en Processing y Wiring (open source).

El Lenguaje de programación Arduino se basa en C/C++, y se simplifica con el uso de la biblioteca Arduino

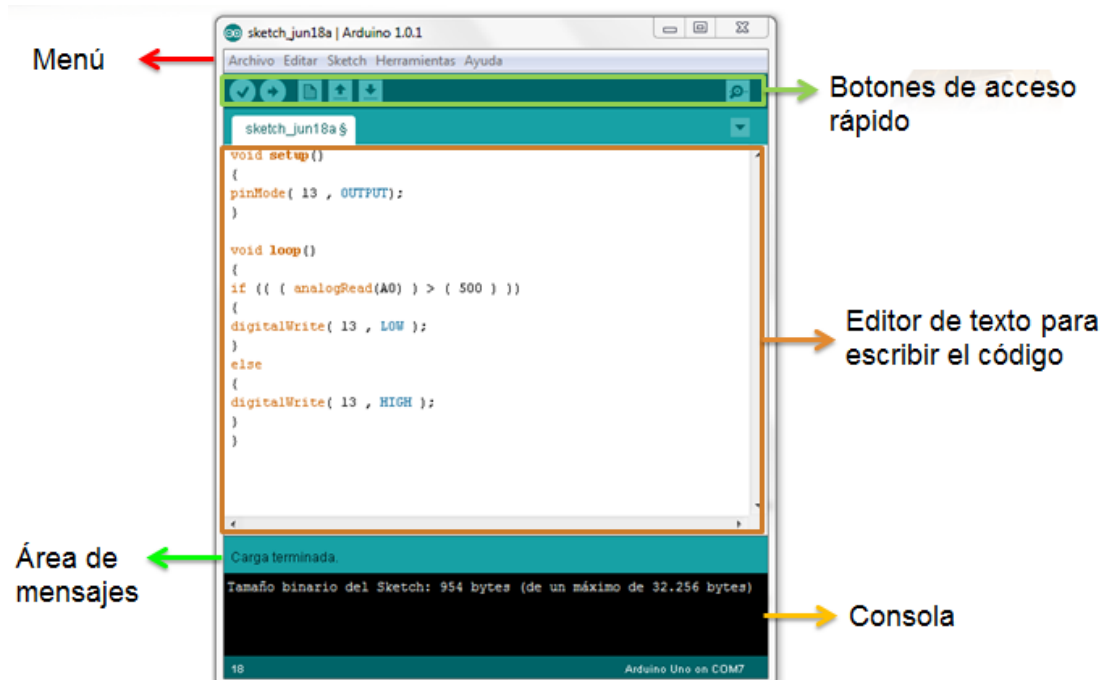


Fig.12 IDE Arduino

Funciones

E/S Digitales

- + pinMode()
- + digitalWrite()
- + digitalRead()

E/S Analógicas

- + analogRead()
- + analogWrite() - PWM
(modulación por ancho de pulso)

Tiempo

- + millis()
- + micros()
- + delay()
- + delayMicroseconds()

Variables

Constantes

- + HIGH | LOW
- + INPUT | OUTPUT
- + true | false
- + Constantes Numéricas

Tipos de Datos

- + boolean (booleano)
- + char (carácter)

Fig.2.8 Comandos en IDE Arduino

2.4.3.- TRANSMISION DE DATOS:

➤ TRANSMISION SERIAL

La comunicación serial es un protocolo muy común (no hay que confundirlo con el Bus Serial de Comunicación, o USB) para comunicación entre dispositivos que se incluye de manera estándar en prácticamente cualquier computadora. La mayoría de las computadoras incluyen dos puertos seriales RS-232. La comunicación serial es también un protocolo común utilizado por varios dispositivos para instrumentación; existen varios dispositivos compatibles con GPIB que incluyen un puerto RS-232. Además, la

comunicación serial puede ser utilizada para adquisición de datos si se usa en conjunto con un dispositivo remoto de muestreo.

El concepto de comunicación serial es sencillo. El puerto serial envía y recibe bytes de información un bit a la vez. Aun y cuando esto es más lento que la comunicación en paralelo, que permite la transmisión de un byte completo por vez, este método de comunicación es más sencillo y puede alcanzar mayores distancias. Por ejemplo, la especificación *IEEE 488* para la comunicación en paralelo determina que el largo del cable para el equipo no puede ser mayor a 20 metros, con no más de 2 metros entre cualesquier dos dispositivos; por el otro lado, utilizando comunicación serial el largo del cable puede llegar a los 1200 metros.

Las características más importantes de la comunicación serial son la velocidad de transmisión, los bits de datos, los bits de parada, y la paridad. Para que dos puertos se puedan comunicar, es necesario que las características sean iguales.

- a. **Velocidad de transmisión (*baud rate*):** Indica el número de bits por segundo que se transfieren, y se mide en baudios (*bauds*).
- b. **Bits de datos:** Se refiere a la cantidad de bits en la transmisión.
- c. **Bits de parada:** Usado para indicar el fin de la comunicación de un solo paquete. Los valores típicos son 1, 1.5 o 2 bits.
- d. **Paridad:** Es una forma sencilla de verificar si hay errores en la transmisión serial. Existen cuatro tipos de paridad: par, impar, marcada y espaciada.

2.4.5 LABVIEW:



Figura 2.9 labview

LabVIEW (acrónimo de Laboratory Virtual Instrumentation Engineering Workbench) es una plataforma y entorno de desarrollo para diseñar sistemas, con un lenguaje de programación visual gráfico. Recomendado para sistemas hardware y software de pruebas, control y diseño, simulado o real y embebido, pues acelera la productividad. El lenguaje que usa se llama lenguaje G, donde la G simboliza que es lenguaje Gráfico.

Como se ha dicho es una herramienta gráfica de programación, esto significa que los programas no se escriben, sino que se dibujan, facilitando su comprensión. Al tener ya pre-diseñados una gran cantidad de bloques, se le facilita al usuario la creación del proyecto, con lo cual

en vez de estar una gran cantidad de tiempo en programar un dispositivo/bloque, se le permite invertir mucho menos tiempo y dedicarse un poco más en la interfaz gráfica y la interacción con el usuario final. Cada VI consta de dos partes diferenciadas:

- *Panel Frontal:* El *Panel Frontal* es la interfaz con el usuario, la utilizamos para interactuar con el usuario cuando el programa se está ejecutando. Los usuarios podrán observar los datos del programa actualizados en tiempo real (como van fluyendo los datos, un ejemplo sería una calculadora, donde tu le pones las entradas, y te pone el resultado en la salida). En esta interfaz se definen los *controles* (los usamos como entradas, pueden ser botones, marcadores etc..) e *indicadores* (los usamos como salidas, pueden ser gráficas).
- *Diagrama de Bloques:* es el programa propiamente dicho, donde se define su funcionalidad, aquí se colocan íconos que realizan una determinada función y se interconectan (el código que controla el programa --. Suele haber una tercera parte *icono/conector* que son los medios utilizados para conectar un VI con otros VIs.--

En el panel frontal, encontraremos todo tipos de controles o indicadores, donde cada uno de estos elementos tiene asignado en el diagrama de bloques una terminal, es decir el usuario podrá diseñar un proyecto en el panel frontal con controles e indicadores, donde estos elementos serán las entradas y salidas que interactuaran con la terminal del VI. Podemos observar en el diagrama de bloques, todos los valores de los controles e indicadores, como van fluyendo entre ellos cuando se está ejecutando un programa VI.

La Figura 1 muestra un *Diagrama de Bloques* de un programa en el que se genera un array de 100 elementos aleatorios, a continuación se hace la FFT de este array y se muestra en una gráfica:

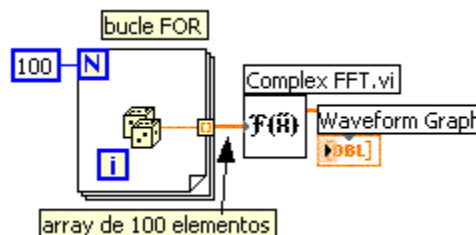


Figura 2.10 Programación labview

3.- MODELAMIENTO DEL BRAZO ROBOT 5GDL:

3.1 ANALISIS CINEMATICO:

3.1.1 CINEMATICA DIRECTA:

El problema cinemático directo se plantea en términos de encontrar una matriz de transformación que relaciona el sistema de coordenadas ligado al cuerpo en movimiento respecto a un sistema de coordenadas que se toma como referencia. Para lograr esta representación se usan las matrices de transformación homogénea 4x4, la cual incluye las operaciones de traslación y la orientación.

La matriz de transformación homogénea es una matriz de 4x4 que transforma un vector expresado en coordenadas homogéneas desde un sistema de coordenadas hasta otro sistema de coordenadas. Para una descripción más amplia acerca de las bases algebraicas de las transformaciones homogéneas se recomienda estudiar las referencias:

La matriz de transformación homogénea tiene la siguiente estructura:

$$T = \begin{bmatrix} \text{matriz de rotación} & \text{vector de posición} \\ f_{1 \times 3} & \text{escalado} \end{bmatrix} = \begin{bmatrix} n_x & s_x & a_x & p_x \\ n_y & s_y & a_y & p_y \\ n_z & s_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T = \begin{bmatrix} n & s & a & p \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

donde los vectores **n**, **s**, **a**, son vectores ortogonales unitarios y **p** es un vector que describe la posición **x**, **y**, **z** del origen del sistema actual respecto del sistema de referencia.

3.2 La representación de Denavit-Hartenberg

La representación de D-H, se aplica a robots de cadena cinemática abierta y consiste en una serie de reglas para colocar los sistemas de referencia de cada eslabón del robot.

Antes de aplicar el método de D-H es importante tener en cuenta los siguientes comentarios:

- Se parte de una configuración cualesquiera del robot, si bien es aconsejable colocarlo en una posición sencilla de analizar.
- Se numeran los eslabones, asignando el 0 para la base y $n-1$ para el último eslabón, siendo n el número de grados de libertad del robot.
- El sistema de coordenadas ortonormal dextrógiro de la base (**x0**, **y0**, **z0**) se establece con el eje **z0** localizado a lo largo del eje de movimiento de la articulación 1 y apuntando hacia fuera del hombro del brazo del robot.

- El sistema de referencia de cada eslabón se coloca al final del mismo, en el extremo de la articulación a la cual esta conectado el eslabón siguiente.
- El ángulo ó desplazamiento de cada eslabón siempre se mide tomando como base el sistema de referencia del eslabón anterior.

Reglas:

- ✓ El eje z_i del sistema de referencia debe quedar alineado a lo largo de la articulación
- ✓ El eje x_i debe colocarse con orientación normal al plano formado por los ejes z_{i-1} y z_i .
- ✓ Al establecer los sistemas de coordenadas de la mano se debe tener en cuenta el principio de Pieper's en el cual se establece que los tres últimos sistemas de referencia se intercepten en un punto, lo cual permite obtener una solución para el problema cinemático inverso de forma cerrada para estas articulaciones.

Además de las reglas anteriores la convención de D-H establece las siguientes condiciones para los demás parámetros geométricos, de acuerdo a la figura-2.2.

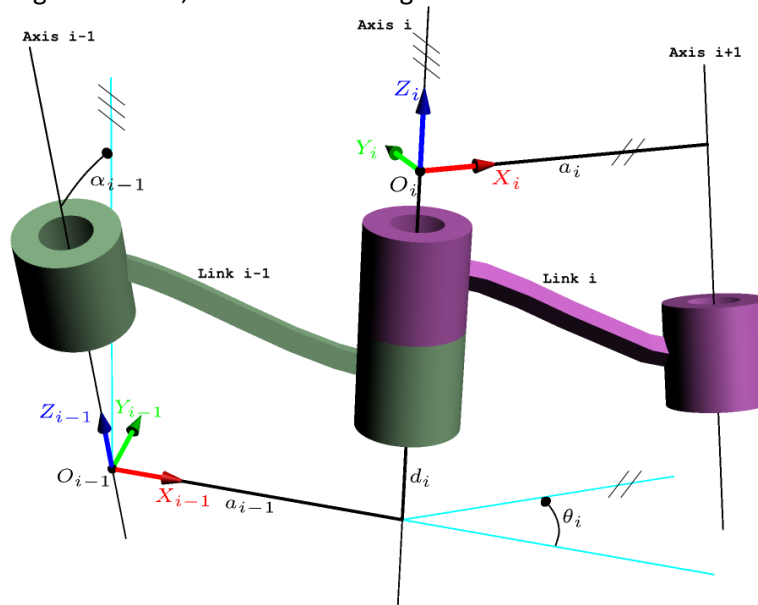


Figura 3.1 Denavit hartemberg

Cada sistema de coordenadas se establece sobre las siguientes reglas.

- θ_i : Es el ángulo de la articulación desde el eje x_{i-1} hasta el eje x_i , medido respecto del eje z_{i-1} , usando la regla de la mano derecha.
- d_i : Es la distancia medida desde el origen del sistema $i-1$, a lo largo del eje z_{i-1} hasta la intersección del eje z_{i-1} con el eje x_i .
- a_i : Es la distancia de separación entre los orígenes de los sistemas de referencia $i-1$ e i , medida a lo largo del eje x_i hasta la intersección con el eje z_{i-1} . (o la distancia más corta entre los ejes z_{i-1} y z_i , cuando estos no se interceptan)
- α_i : Es el ángulo que separa los ejes z_i y z_{i-1} , medido respecto del eje x_i

$${}^{i-1}A_i = \begin{bmatrix} c\theta_i & -c\alpha_i s\theta_i & s\alpha_i s\theta_i & a_i c\theta_i \\ s\theta_i & c\alpha_i c\theta_i & -s\alpha_i c\theta_i & a_i s\theta_i \\ 0 & s\alpha_i & c\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Los cálculos se desarrollaron en Matlab. Para el análisis de cinemática Directa del Robot manipulador de 5GDL. Se presentan a continuación:

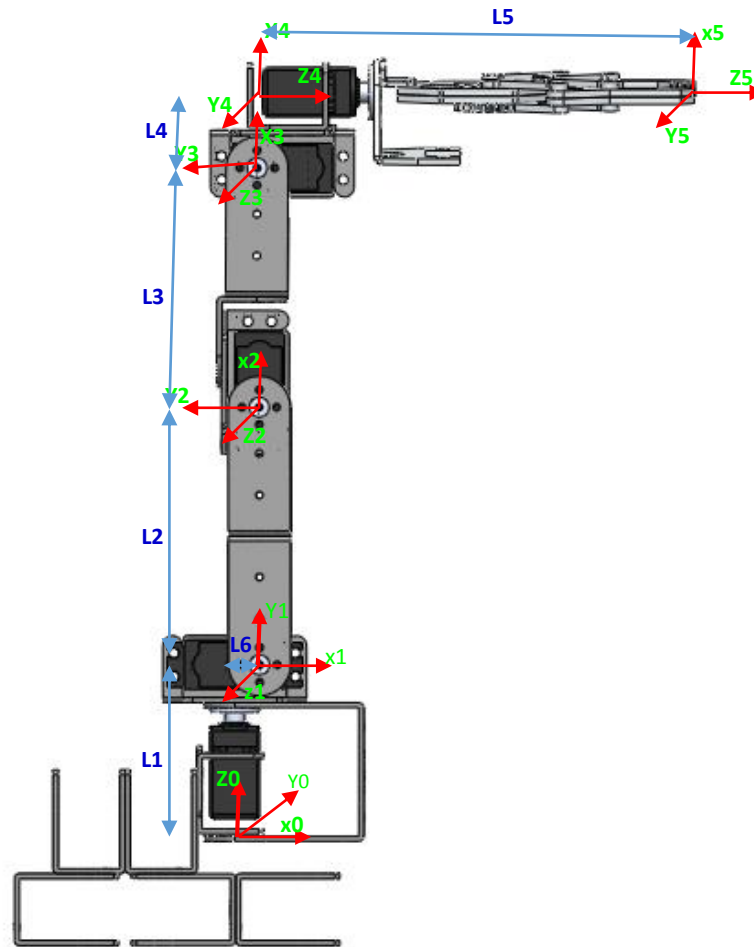


Figura 3.2 Sistema de referencia Denavit – Hartember.

Siendo:

- L1= 70mm
- L2= 105mm
- L3= 97mm
- L4= 30mm
- L5= 160mm
- L6= 10mm

```
% Análisis CINEMATICO de UN Brazo Robot de 5GDL.
% CURSO: Robotica Industrial -UNT
% Fecha: 14/09/2015
% ELABORADO BY JOSMELL ALVA ALCANTARA.
%-----
clc,clear all
syms q1 q2 q3 q4 q5 L1 L2 L3 L4 L5 L6 Px Py Pz;
syms pi;

% CINEMATICA DIRECTA
%Parametros de Denavit H:
%VALORES NUMERICOS L1=70mm; L2=105mm; L3=97mm; L4=30mm; L5=160mm L6=10mm;
theta=[q1; q2+pi/2; q3; q4; q5];
d=[L1; 0; 0; 0; L5];
alpha=[pi/2; 0; 0 ;pi/2; 0];
a=[L6; L2; L3;L4; 0];
MDH=[theta d a alpha] % muestra la tabla de matriz de transformacion
Homogenea

% Matrices de transformacion Homogenea.
T01=denavit(theta(1),d(1),a(1),alpha(1))
T12=denavit(theta(2),d(2),a(2),alpha(2))
T23=denavit(theta(3),d(3),a(3),alpha(3))
T34=denavit(theta(4),d(4),a(4),alpha(4))
T45=denavit(theta(5),d(5),a(5),alpha(5))

%Posicion Final con respecto a la Base
T05=simplify(T01*T12*T23*T34*T45)
```

Los parametros de Denavit – Hartenberg

$$\begin{matrix}
 \text{MDH} = \\
 \begin{bmatrix}
 q_1, & L_1, & L_6, & \pi/2 \\
 \pi/2 + q_2, & 0, & L_2, & 0 \\
 q_3, & 0, & L_3, & 0 \\
 q_4, & 0, & L_4, & \pi/2 \\
 q_5, & L_5, & 0, & 0
 \end{bmatrix}
 \end{matrix}
 \quad (3.1)$$

T01 =

$$\begin{bmatrix}
 \cos(q_1), 0, \sin(q_1), L_6 \cos(q_1) \\
 \sin(q_1), 0, -\cos(q_1), L_6 \sin(q_1) \\
 0, 1, 0, L_1 \\
 0, 0, 0, 1
 \end{bmatrix}
 \quad (3.2)$$

T12 =

$$\begin{bmatrix}
 \cos(\pi/2 + q_2), -\sin(\pi/2 + q_2), 0, L_2 \cos(\pi/2 + q_2) \\
 \sin(\pi/2 + q_2), \cos(\pi/2 + q_2), 0, L_2 \sin(\pi/2 + q_2)
 \end{bmatrix}
 \quad (3.3)$$

$$\begin{bmatrix} 0, & 0, 1, & 0 \\ 0, & 0, 0, & 1 \end{bmatrix}$$

T23 =

$$\begin{bmatrix} \cos(q_3), & -\sin(q_3), & 0, & L_3 \cos(q_3) \\ \sin(q_3), & \cos(q_3), & 0, & L_3 \sin(q_3) \\ 0, & 0, 1, & 0 \\ 0, & 0, 0, & 1 \end{bmatrix} \quad (3.4)$$

T34 =

$$\begin{bmatrix} \cos(q_4), & 0, & \sin(q_4), & L_4 \cos(q_4) \\ \sin(q_4), & 0, & -\cos(q_4), & L_4 \sin(q_4) \\ 0, & 1, & 0, & 0 \\ 0, & 0, & 0, & 1 \end{bmatrix} \quad (3.5)$$

T45 =

$$\begin{bmatrix} \cos(q_5), & -\sin(q_5), & 0, & 0 \\ \sin(q_5), & \cos(q_5), & 0, & 0 \\ 0, & 0, 1, & L_5 \\ 0, & 0, 0, & 1 \end{bmatrix} \quad (3.6)$$

$$T_{05} = T_{01} * T_{12} * T_{23} * T_{34} * T_{45}$$

$$T_{05} = \begin{pmatrix} r_{11} & r_{12} & r_{13} & P_x \\ r_{21} & r_{22} & r_{23} & P_y \\ r_{31} & r_{32} & r_{33} & P_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (3.7)$$

$$r_{11} = \sin(q_1) * \sin(q_5) - (\cos(q_5) * (\sin(q_1 + q_2 + q_3 + q_4) - \sin(q_1 - q_2 - q_3 - q_4))) / 2;$$

$$r_{21} = \cos(q_5) * (\cos(q_1 + q_2 + q_3 + q_4) / 2 - \cos(q_1 - q_2 - q_3 - q_4) / 2) - \cos(q_1) * \sin(q_5);$$

$$r_{31} = \cos(q_2 + q_3 + q_4 + q_5) / 2 + \cos(q_2 + q_3 + q_4 - q_5) / 2;$$

$$r_{12} = \cos(q_5) * \sin(q_1) + \sin(q_5) * (\sin(q_1 + q_2 + q_3 + q_4) / 2 - \sin(q_1 - q_2 - q_3 - q_4) / 2);$$

$$r_{22} = -\cos(q_1) * \cos(q_5) - (\sin(q_5) * (\cos(q_1 + q_2 + q_3 + q_4) - \cos(q_1 - q_2 - q_3 - q_4))) / 2;$$

$$r_{32} = \cos(q_2 + q_3 + q_4 + q_5)/2 + \cos(q_2 + q_3 + q_4 - q_5)/2;$$

$$r_{13} = \cos(q_1 + q_2 + q_3 + q_4)/2 + \cos(q_1 - q_2 - q_3 - q_4)/2;$$

$$r_{23} = \sin(q_1 + q_2 + q_3 + q_4)/2 + \sin(q_1 - q_2 - q_3 - q_4)/2;$$

$$r_{33} = \sin(q_2 + q_3 + q_4);$$

$$P_x = (L_5 \cdot \cos(q_1 + q_2 + q_3 + q_4))/2 - (L_4 \cdot \sin(q_1 + q_2 + q_3 + q_4))/2 + (L_5 \cdot \cos(q_1 - q_2 - q_3 - q_4))/2 - (L_2 \cdot \sin(q_1 + q_2))/2 + (L_4 \cdot \sin(q_1 - q_2 - q_3 - q_4))/2 + L_6 \cdot \cos(q_1) + (L_3 \cdot \sin(q_1 - q_2 - q_3))/2 + (L_2 \cdot \sin(q_1 - q_2))/2 - (L_3 \cdot \sin(q_1 + q_2 + q_3))/2;$$

$$P_y = (L_4 \cdot \cos(q_1 + q_2 + q_3 + q_4))/2 + (L_5 \cdot \sin(q_1 + q_2 + q_3 + q_4))/2 + (L_2 \cdot \cos(q_1 + q_2))/2 - (L_4 \cdot \cos(q_1 - q_2 - q_3 - q_4))/2 + (L_5 \cdot \sin(q_1 - q_2 - q_3 - q_4))/2 - (L_3 \cdot \cos(q_1 - q_2 - q_3))/2 + L_6 \cdot \sin(q_1) - (L_2 \cdot \cos(q_1 - q_2))/2 + (L_3 \cdot \cos(q_1 + q_2 + q_3))/2;$$

$$P_z = L_1 + L_3 \cdot \cos(q_2 + q_3) + L_2 \cdot \cos(q_2) + L_4 \cdot \cos(q_2 + q_3 + q_4) + L_5 \cdot \sin(q_2 + q_3 + q_4);$$

3.1.2 CIEMATICA INVERSA:

La cinemática inversa consiste en hallar los valores de las coordenadas articulares del robot : $q_1, q_2, q_3, \dots, q_n$ donde $n = 1, 2$ conocida la posición y orientación del extremo del robot.

Se pueden encontrar diversos métodos genéricos para la resolución de la cinemática inversa que pueden ser implementados en computadora, suele ser habitual la resolución por medio de métodos geométricos.

La mayor parte de los robots suelen tener cadenas cinemáticas relativamente sencillas, que facilitan la utilización de los métodos geométricos. Para muchos robots, si se consideran sólo los tres primeros grados de libertad, se tiene una estructura plana. Este hecho facilita la resolución del problema. Asimismo los últimos tres grados de libertad suelen usarse para la orientación de la herramienta, lo cual permite una resolución geométrica desacoplada de la posición de la muñeca del robot y de la orientación de la herramienta.

En esta sección se va a resolver el problema cinemático inverso para los dos robots anteriores, utilizando el método geométrico e implementándolo en **Matlab**.

Mediante el método de desacoplo cinemático, obtuvimos las ecuaciones para calcular los ángulos, Asumiendo por ahora que el ángulo q_5 siempre será cero.

Calculamos los parámetros necesarios:

$$K1 = (P03z - L1) ;$$

$$K2 = (P03x + P03y * \sin(q1) - d1) ;$$

$$a = K2 * L2 ;$$

$$b = 2 * K1 * L2 ;$$

$$c = -L3 + K1^2 + K2^2 + L2^2 ;$$

Ahora Calculamos los ángulos

Angulo q1:

$$q1 = \text{atan}(P03y, P03x)$$

Ángulo q2:

$$q2_1 = 2 * \text{atan}((b + \sqrt{b^2 + a^2 - c^2}) / (a + c))$$

$$q2_2 = 2 * \text{atan}((b - \sqrt{b^2 + a^2 - c^2}) / (a + c))$$

Ángulo q3:

$$q3 = \text{atan}(\sin q3, \cos q3)$$

Ángulo q4:

$$\sin q4 = (\sin(q2 + q3) * \cos(q1)) a05x + (\sin(q2 + q3) * \sin(q1)) * a05y + \cos(q2 + q3) * a05z$$

$$\cos q4 = \sqrt{1 - \sin q4^2} ;$$

$$q4 = \text{atan}(\sin q4, \cos q4) ;$$

Ángulo q5

$$q5 = 0 ;$$

3.3 ANALISIS DINAMICO:

La dinámica del robot relaciona el movimiento del robot y las fuerzas implicadas en el mismo. El modelo dinámico establece relaciones matemáticas entre las coordenadas articulares (o las coordenadas del extremo del robot), sus derivadas (velocidad y aceleración), las fuerzas y pares aplicados en las articulaciones (o en el extremo) y los parámetros del robot (masas de los eslabones, inercias, etc).

Siguiendo con la filosofía de este libro, se recomienda al lector que quiera profundizar sobre la dinámica de robots, la lectura de los textos [1],[2],[3], donde se estudian varias formulaciones clásicas como Lagrange-Euler o las ecuaciones generalizadas de D'Alembert. Hay que tener en cuenta que las ecuaciones de movimiento obtenidas con estas formulaciones son equivalentes en el sentido que describen la conducta dinámica del robot, sin embargo, cada una de ellas presenta

características diferentes que la hacen más apropiada para ciertas tareas.

Por ejemplo, la formulación de Lagrange-Euler presenta un modelo simple y elegante, dando como resultado una serie de ecuaciones diferenciales no lineales de 2º orden acopladas útiles para el estudio de estrategias de control en el espacio de estados de las variables articulares del robot, pero que se presentan ineficaces para aplicaciones en tiempo real dado el elevado tiempo de computación que requieren las operaciones con matrices de transformación homogénea.

Los modelos dinámicos que se estudian en esta práctica están basados en el algoritmo recursivo de Newton-Euler (N-E) desarrollado por Luh [1]. Aunque las formulaciones recursivas destruyen la estructura del modelo dinámico analítico y dan lugar a la falta de ecuaciones cerradas necesarias para el análisis del control, la dificultad de un análisis clásico es enorme debido a que se obtienen expresiones fuertemente no-lineales que constan de cargas inerciales, fuerzas de acoplo entre las articulaciones y efectos de las cargas de gravedad, con la dificultad añadida de que los pares/fuerzas dinámicos dependen de los parámetros físicos del manipulador, de la configuración instantánea de las articulaciones, de la velocidad, de la aceleración y de la carga que soporta el robot.

Aunque las ecuaciones del movimiento son equivalentes ya sean analíticas o recursivas, los diferentes planteamientos dependen de los objetivos que se quieran conseguir con ellos. En algunos casos es necesario solucionar el problema dinámico de un robot para lograr tiempos de cálculo rápidos en la evaluación de los pares y fuerzas articulares para controlar el manipulador, y en otros casos son necesarios planteamientos para facilitar el análisis y la síntesis del control.

3.2.-Dinámica inversa. La formulación de Newton-Euler:

El método de Newton-Euler permite obtener un conjunto de ecuaciones recursivas hacia delante de velocidad y aceleración lineal y angular las cuales están referidas a cada sistema de referencia articular. Las velocidades y aceleraciones de cada elemento se propagan hacia adelante desde el sistema de referencia de la base hasta el efector final. Las ecuaciones recursivas hacia atrás calculan los pares y fuerzas necesarios para cada articulación desde la mano (incluyendo en ella efectos de fuerzas externas), hasta el sistema de referencia de la base.

3.2.1. Sistemas de coordenadas en movimiento:

La formulación de N-E se basa en los sistemas de coordenadas en movimiento

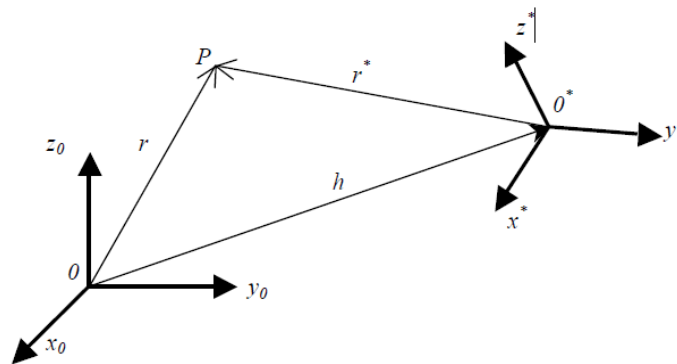


Figura 3.3 Sistema de Coordenadas de Movimiento

3.2.3. Ecuaciones de movimiento recursivas.

A partir de las ecuaciones cinemáticas del apartado anterior y aplicando el principio de D'Alembert del equilibrio estático para todos los instantes de tiempo, se obtienen las ecuaciones recursivas de Newton-Euler.[1]

Si se utiliza la nomenclatura de la figura 3.2 sobre un eslabón cualquiera del robot, tal y como se muestra en la figura 3.3

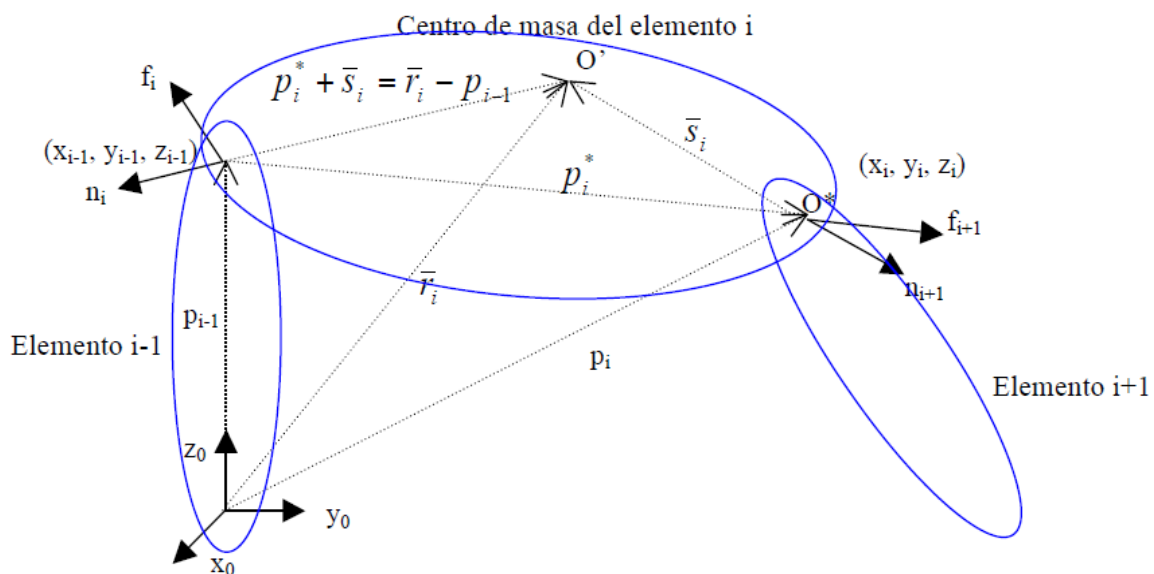


Figura 3.4 Dinámica newton Euler Sistema de referencia

Donde:

M_i : masa total del eslabón i ,

R_i : posición del centro de masa del elemento i desde el origen del sistema de referencia de la base,

S_i : posición del centro de masa del elemento i desde el origen del sistema de coordenadas (x_i, y_i, z_i) ,
 P_i : posición del origen de coordenadas i -ésimo con respecto al sistema de coordenadas $(i-1)$ -ésimo,
 v_i = velocidad lineal del centro de masa del elemento i ,
 a_i = aceleración lineal del centro de masa del elemento i ,
 F_i = fuerza total externa ejercida sobre el elemento i en el centro de masa,
 N_i = momento total externo ejercido sobre el elemento i en el centro de masa,
 I_i = matriz de inercia del elemento i respecto de su centro de masa con respecto al sistema de coordenadas (x_0, y_0, z_0) ,

Se desarrolló los algoritmos en Matlab para el cálculo de la dinámica y se obtienen los siguientes resultados:

```
%DINAMICA DE BRAZO ROBOT 5GDL - ROBOTICA INDUSTRIAL
% ING. MECATRONICA - UNIVERSIDAD NACIONAL DE TRUJILLO
% ELABORADO BY JOSMELL ALVA ALCANTARA
% FECHA: 03/12/2015
```

```
%-----
--
```

```
clc, clear all
syms q1 q2 q3 q4 q5 L1 L2 L3 L4 L5 L6 m1 m2 m3 m4 m5
syms dq1 dq2 dq3 dq4 dq5 ddq1 ddq2 ddq3 ddq4 ddq5
syms pi g Mext Iext I1 I2 I3 I4 I5 b1 b2 b3 b4 b5
```

```
%-----
%parametro de Denavit-Hartenberg del robot
%-----
theta=[q1; q2+pi/2; q3; q4; q5];
d=[L1; 0; 0; 0; L5];
a=[L6; L2; L3; L4; 0];
alpha=[pi/2; 0; 0; pi/2; 0];
DH=[theta d a alpha]
```

```
%-----
% Masa de cada elemento en Kg
%-----
% m1=elemento1
% m2=elemento2
```

```
% -----
% Coeficiente de rozamiento viscoso de cada articulacion
% -----
% b1=Coeficiente viscoso elemento1
% b2=Coeficiente viscoso elemento2
```

```
%-----
% Matrices de Inercia (Kg-m^2)
```

```

% -----
r10I_r01=I1;
r20I_r02=I2;
r30I_r03=I3;
r40I_r04=I4;
r50I_r05=I5;

% -----
% Matrices de Rotación.
% -----
r01=denavit(theta(1),d(1),a(1),alpha(1));
r01=r01(1:3,1:3)
r10=r01.'
r12=denavit(theta(2),d(2),a(2),alpha(2));
r12=r12(1:3,1:3)
r21=r12.'
r23=denavit(theta(3),d(3),a(3),alpha(3));
r23=r23(1:3,1:3)
r32=r23.'
r34=denavit(theta(4),d(4),a(4),alpha(4));
r34=r34(1:3,1:3)
r43=r34.'
r45=denavit(theta(5),d(5),a(5),alpha(5));
r45=r45(1:3,1:3)
r54=r45.'
r56=eye(3);
r65=r56';

% -----
% Vectores ri0pi, ri0si.
% -----
r10p1 = ri0pi(a(1),d(1),alpha(1))
r20p2 = ri0pi(a(2),d(2),alpha(2))
r30p3 = ri0pi(a(3),d(3),alpha(3))
r40p4 = ri0pi(a(4),d(4),alpha(4))
r50p5 = ri0pi(a(5),d(5),alpha(5))
r60p6=zeros(3,1);

%posicion del centro de masa del elemento i desde sistema de referencia i
base
r10s1 = [0;-L1/2;0]
r20s2 = [-L2/2;0;0]
r30s3 = [-L3/2;0;0]
r40s4 = [-L4/2;0;0]
r50s5 = [0;0;L5/2]
r60s6 = zeros(3,1)

% -----
% Velocidad angular de las articulaciones
% -----
r00w0=zeros(3,1); % velocidad inicial =0
r10w1=ri0wi(r10,r00w0,dq1)
r20w2=ri0wi(r21,r10w1,dq2)
r30w3=ri0wi(r32,r20w2,dq3)
r40w4=ri0wi(r43,r30w3,dq4)
r50w5=ri0wi(r54,r40w4,dq5)

```

```

r60w6=ri0wi(r65,r50w5,0)

% -----
% Aceleracion angular de las articulaciones
% -----
r00wp0 = zeros(3,1); % ACELERACION INICIAAL.
r10wp1 = ri0wpi(r10, r00wp0,r00w0,dq1,ddq1)
r20wp2 = ri0wpi(r21, r10wp1,r10w1,dq2,ddq2)
r30wp3 = ri0wpi(r32, r20wp2,r20w2,dq3,ddq3)
r40wp4 = ri0wpi(r43, r30wp3,r30w3,dq4,ddq4)
r50wp5 = ri0wpi(r54, r40wp4,r40w4,dq5,ddq5)
r60wp6 = ri0wpi(r65, r50wp5,r50w5,0,0)

% -----
% Aceleracion lineal articular
% -----
r00vp0 = [-g; 0; 0] % aceleracion de la gravedad
r10vp1 = simplify(ri0vpi_r(r10, r00vp0, r10wp1, r10w1, r10p1))
r20vp2 = simplify(ri0vpi_r(r21, r10vp1, r20wp2, r20w2, r20p2))
r30vp3 = simplify(ri0vpi_r(r32, r20vp2, r30wp3, r30w3, r30p3))
r40vp4 = simplify(ri0vpi_r(r43, r30vp3, r40wp4, r40w4, r40p4))
r50vp5 = simplify(ri0vpi_r(r54, r40vp4, r50wp5, r50w5, r50p5))

r60vp6 = simplify(ri0vpi_r(r65, r50vp5, r60wp6, r60w6, r60p6));

% -----
% Aceleracion del centro de masa de cada elemento
% -----
r10a1 = simplify(ri0ai(r10vp1, r10wp1, r10w1, r10s1))
r20a2 = simplify(ri0ai(r20vp2, r20wp2, r20w2, r20s2))
r30a3 = simplify(ri0ai(r30vp3, r30wp3, r30w3, r30s3))
r40a4 = simplify(ri0ai(r40vp4, r40wp4, r40w4, r40s4))
r50a5 = simplify(ri0ai(r50vp5, r50wp5, r50w5, r50s5))
r60a6 = simplify(ri0ai(r60vp6, r60wp6, r60w6, r60s6));

% -----
% Fuerza en el centro de masa de cada elemento
% -----

r60f6 = ri0fi(r60a6, 0) % fuerza inicial de carga =0
r50f5 = ri0fi(r50a5, m5)
r40f4 = ri0fi(r40a4, m4)
r30f3 = ri0fi(r30a3, m3)
r20f2 = ri0fi(r20a2, m2)
r10f1 = ri0fi(r10a1, m1)

% -----
% Par en el centro de masa de cada elemento
% -----

r60n6 = ri0ni(r60wp6, r60w6, Iext);
r50n5 = ri0ni(r50wp5, r50w5, r50I_r05)
r40n4 = ri0ni(r40wp4, r40w4, r40I_r04)
r30n3 = ri0ni(r30wp3, r30w3, r30I_r03)
r20n2 = ri0ni(r20wp2, r20w2, r20I_r02)

```

```

r10n1 = ri0ni(r10wp1, r10w1, r10I_r01)

% -----
% Fuerzas articulares
% -----
r60f6a = r60f6
r50f5a = simplify(ri0fia(r56, r60f6a, r50f5))
r40f4a = simplify(ri0fia(r45, r50f5a, r40f4))
r30f3a = simplify(ri0fia(r34, r40f4a, r30f3))
r20f2a = simplify(ri0fia(r23, r30f3a, r20f2))
r10f1a = simplify(ri0fia(r12, r20f2a, r10f1))

% -----
% Pares articulares
% -----

r20p1 = r21*(r10p1);
r30p2 = r32*(r20p2);
r40p3 = r43*(r30p3);
r50p4 = r54*(r40p4);
r60p5 = r65*(r50p5);

r60n6a = r60n6
r50n5a =simplify(ri0nia(r56, r60n6a, r60f6a, r50n5, r50f5, r60p5,
r50p5,r50s5))
r40n4a =simplify(ri0nia(r45, r50n5a, r50f5a, r40n4, r40f4, r50p4,
r40p4,r40s4))
r30n3a =simplify(ri0nia(r34, r40n4a, r40f4a, r30n3, r30f3, r40p3,
r30p3,r30s3))
r20n2a =simplify(ri0nia(r23, r30n3a, r30f3a, r20n2, r20f2, r30p2,
r20p2,r20s2))
r10n1a =simplify(ri0nia(r12, r20n2a, r20f2a, r10n1, r10f1, r20p1,
r10p1,r10s1))

% -----
% Fuerzas y pares de accionamientos
% -----
t_1 = simplify(t_r(r10, r10n1a, dq1, b1))
t_2 = simplify(t_r(r21, r20n2a, dq2, b2))
t_3 = simplify(t_r(r32, r30n3a, dq3, b3))
t_4 = simplify(t_r(r43, r40n4a, dq4, b4))
t_5 = simplify(t_r(r54, r50n5a, dq5, b5))
tau = [t_1; t_2 ; t_3; t_4; t_5]

```

Siendo:

q_1, q_2, q_3, q_4, q_5 : Posiciones articulares:

$\dot{q}_1, \dot{q}_2, \dot{q}_3, \dot{q}_4, \dot{q}_5$: velocidades articulares.

$\ddot{q}_1, \ddot{q}_2, \ddot{q}_3, \ddot{q}_4, \ddot{q}_5$: Aceleraciones articulares

I_1, I_2, I_3, I_4, I_5 : Momentos de inercia de cada articulación.

I_{ext}, m_{ext} : Momento de inercia y masa externa (soporta en la pinza del brazo robot 5gdl).

$L_1, L_2, L_3, L_4, L_5, L_6$: Longitud de cada eslabón del robot Manipulador

Los resultados Obtenidos son los Siguietes.

DINAMICA NEWTON EULER- BRAZO ROBOT 5GDL

$$\tau_5 = b_5 \dot{q}_5 + (I_5 + I_{ext}) (\ddot{q}_5 + \ddot{q}_1 \sin(q_2 + q_3 + q_4) + \dot{q}_1 \dot{q}_2 \cos(q_2 + q_3 + q_4) + \dot{q}_1 \dot{q}_3 \cos(q_2 + q_3 + q_4) + \dot{q}_1 \dot{q}_4 \cos(q_2 + q_3 + q_4))$$

$$\begin{aligned} \tau_4 = & I_4 \ddot{q}_2 + I_4 \ddot{q}_3 + I_5 \ddot{q}_2 + I_4 \ddot{q}_4 + I_5 \ddot{q}_3 + I_5 \ddot{q}_4 + I_{ext} \ddot{q}_2 + I_{ext} \ddot{q}_3 + I_{ext} \ddot{q}_4 + b_4 \dot{q}_4 + (L_4^2 \ddot{q}_2^2 m_4)/4 + L_4^2 \ddot{q}_2^2 m_5 + (L_4^2 \ddot{q}_3^2 m_4)/4 + \\ & L_4^2 \ddot{q}_3^2 m_5 + (L_4^2 \ddot{q}_4^2 m_4)/4 + (9 L_5^2 \ddot{q}_2^2 m_5)/4 + L_4^2 \ddot{q}_4^2 m_5 + (9 L_5^2 \ddot{q}_3^2 m_5)/4 + (9 L_5^2 \ddot{q}_4^2 m_5)/4 + (L_4^2 g m_4 \cos(q_1 + q_2 + q_3 + q_4))/4 + \\ & (L_4^2 g m_5 \cos(q_1 + q_2 + q_3 + q_4))/2 + (3 L_5^2 g m_5 \sin(q_1 + q_2 + q_3 + q_4))/4 + (L_4^2 g m_4 \cos(q_1 - q_2 - q_3 - q_4))/4 + (L_4^2 g m_5 \cos(q_1 - q_2 - q_3 - q_4))/2 - \\ & (3 L_5^2 g m_5 \sin(q_1 - q_2 - q_3 - q_4))/4 - (L_4^2 \dot{q}_1^2 m_4 \sin(2 q_2 + 2 q_3 + 2 q_4))/8 - (L_4^2 \dot{q}_1^2 m_5 \sin(2 q_2 + 2 q_3 + 2 q_4))/2 + (9 L_5^2 \dot{q}_1^2 m_5 \sin(2 q_2 + 2 q_3 + 2 q_4))/8 - \\ & I_5 \dot{q}_1 \dot{q}_5 \cos(q_2 + q_3 + q_4) - I_{ext} \dot{q}_1 \dot{q}_5 \cos(q_2 + q_3 + q_4) + (L_3 L_4 \dot{q}_1^2 m_4 \sin(q_4))/4 + (L_3 L_4 \dot{q}_1^2 m_5 \sin(q_4))/2 + \\ & (L_3 L_4 \dot{q}_2^2 m_4 \sin(q_4))/2 + L_3 L_4 \dot{q}_2^2 m_5 \sin(q_4) + (L_3 L_4 \dot{q}_3^2 m_4 \sin(q_4))/2 + L_3 L_4 \dot{q}_3^2 m_5 \sin(q_4) - (L_3 L_4 \dot{q}_1^2 m_4 \sin(2 q_2 + 2 q_3 + q_4))/4 - \\ & (L_3 L_4 \dot{q}_1^2 m_5 \sin(2 q_2 + 2 q_3 + q_4))/2 + (L_4 L_6 \dot{q}_1^2 m_4 \cos(q_2 + q_3 + q_4))/2 + L_4 L_6 \dot{q}_1^2 m_5 \cos(q_2 + q_3 + q_4) + (3 L_5 L_6 \dot{q}_1^2 m_5 \sin(q_2 + q_3 + q_4))/2 + \\ & (3 L_4 L_5 \dot{q}_1^2 m_5 \cos(2 q_2 + 2 q_3 + 2 q_4))/2 + (L_2 L_4 \ddot{q}_2^2 m_4 \cos(q_3 + q_4))/2 + L_2 L_4 \ddot{q}_2^2 m_5 \cos(q_3 + q_4) + (3 L_2 L_5 \ddot{q}_2^2 m_5 \sin(q_3 + q_4))/2 + \\ & (L_3 L_4 \ddot{q}_2^2 m_4 \cos(q_4))/2 + L_3 L_4 \ddot{q}_2^2 m_5 \cos(q_4) + (L_3 L_4 \ddot{q}_3^2 m_4 \cos(q_4))/2 + L_3 L_4 \ddot{q}_3^2 m_5 \cos(q_4) + (3 L_2 L_5 \dot{q}_1^2 m_5 \cos(2 q_2 + q_3 + q_4))/4 + \end{aligned}$$

$$\begin{aligned}
& (3*L3*L5*ddq2*m5*\sin(q4))/2 + (3*L3*L5*ddq3*m5*\sin(q4))/2 - \\
& (L2*L4*dq1^2*m4*\sin(2*q2 + q3 + q4))/4 - (L2*L4*dq1^2*m5*\sin(2*q2 + q3 + q4))/2 - \\
& (3*L2*L5*dq1^2*m5*\cos(q3 + q4))/4 - (3*L2*L5*dq2^2*m5*\cos(q3 + q4))/2 + \\
& (L2*L4*dq1^2*m4*\sin(q3 + q4))/4 + (L2*L4*dq1^2*m5*\sin(q3 + q4))/2 + \\
& (L2*L4*dq2^2*m4*\sin(q3 + q4))/2 + L2*L4*dq2^2*m5*\sin(q3 + q4) - \\
& (3*L3*L5*dq1^2*m5*\cos(q4))/4 - (3*L3*L5*dq2^2*m5*\cos(q4))/2 - \\
& (3*L3*L5*dq3^2*m5*\cos(q4))/2 + (3*L3*L5*dq1^2*m5*\cos(2*q2 + 2*q3 + q4))/4 - \\
& 3*L3*L5*dq2*dq3*m5*\cos(q4) + L3*L4*dq2*dq3*m4*\sin(q4) + \\
& 2*L3*L4*dq2*dq3*m5*\sin(q4)
\end{aligned}$$

$$\begin{aligned}
& \tau = \\
& I3*ddq2 + I3*ddq3 + I4*ddq2 + I4*ddq3 + I5*ddq2 + I4*ddq4 + I5*ddq3 + I5*ddq4 + \\
& Iext*ddq2 + Iext*ddq3 + Iext*ddq4 + b3*dq3 + (L3^2*ddq2*m3)/4 + L3^2*ddq2*m4 + \\
& (L3^2*ddq3*m3)/4 + L3^2*ddq2*m5 + L3^2*ddq3*m4 + (L4^2*ddq2*m4)/4 + \\
& L3^2*ddq3*m5 + L4^2*ddq2*m5 + (L4^2*ddq3*m4)/4 + L4^2*ddq3*m5 + \\
& (L4^2*ddq4*m4)/4 + (9*L5^2*ddq2*m5)/4 + L4^2*ddq4*m5 + (9*L5^2*ddq3*m5)/4 + \\
& (9*L5^2*ddq4*m5)/4 + (L4*g*m4*\cos(q1 + q2 + q3 + q4))/4 + (L4*g*m5*\cos(q1 + q2 + q3 \\
& + q4))/2 + (3*L5*g*m5*\sin(q1 + q2 + q3 + q4))/4 + (L4*g*m4*\cos(q1 - q2 - q3 - q4))/4 + \\
& (L4*g*m5*\cos(q1 - q2 - q3 - q4))/2 - (3*L5*g*m5*\sin(q1 - q2 - q3 - q4))/4 + \\
& (L3*g*m3*\cos(q1 - q2 - q3))/4 + (L3*g*m4*\cos(q1 - q2 - q3))/2 + (L3*g*m5*\cos(q1 - q2 - \\
& q3))/2 - (L4^2*dq1^2*m4*\sin(2*q2 + 2*q3 + 2*q4))/8 - (L4^2*dq1^2*m5*\sin(2*q2 + 2*q3 \\
& + 2*q4))/2 + (9*L5^2*dq1^2*m5*\sin(2*q2 + 2*q3 + 2*q4))/8 - I5*dq1*dq5*\cos(q2 + q3 + \\
& q4) - Iext*dq1*dq5*\cos(q2 + q3 + q4) + (L3*g*m3*\cos(q1 + q2 + q3))/4 + (L3*g*m4*\cos(q1 \\
& + q2 + q3))/2 + (L3*g*m5*\cos(q1 + q2 + q3))/2 - (L3^2*dq1^2*m3*\sin(2*q2 + 2*q3))/8 - \\
& (L3^2*dq1^2*m4*\sin(2*q2 + 2*q3))/2 - (L3^2*dq1^2*m5*\sin(2*q2 + 2*q3))/2 + \\
& (L2*L3*dq1^2*m3*\sin(q3))/4 + (L2*L3*dq1^2*m4*\sin(q3))/2 + \\
& (L2*L3*dq2^2*m3*\sin(q3))/2 + (L2*L3*dq1^2*m5*\sin(q3))/2 + L2*L3*dq2^2*m4*\sin(q3) \\
& + L2*L3*dq2^2*m5*\sin(q3) - (L3*L4*dq4^2*m4*\sin(q4))/2 - L3*L4*dq4^2*m5*\sin(q4) - \\
& (L3*L4*dq1^2*m4*\sin(2*q2 + 2*q3 + q4))/2 - L3*L4*dq1^2*m5*\sin(2*q2 + 2*q3 + q4) - \\
& (L2*L3*dq1^2*m3*\sin(2*q2 + q3))/4 - (L2*L3*dq1^2*m4*\sin(2*q2 + q3))/2 - \\
& (L2*L3*dq1^2*m5*\sin(2*q2 + q3))/2 + (L4*L6*dq1^2*m4*\cos(q2 + q3 + q4))/2 + \\
& L4*L6*dq1^2*m5*\cos(q2 + q3 + q4) + (3*L5*L6*dq1^2*m5*\sin(q2 + q3 + q4))/2 + \\
& (3*L4*L5*dq1^2*m5*\cos(2*q2 + 2*q3 + 2*q4))/2 + (L2*L4*ddq2*m4*\cos(q3 + q4))/2 + \\
& L2*L4*ddq2*m5*\cos(q3 + q4) + (3*L2*L5*ddq2*m5*\sin(q3 + q4))/2 +
\end{aligned}$$

$$\begin{aligned}
& (L2*L3*ddq2*m3*cos(q3))/2 + L2*L3*ddq2*m4*cos(q3) + L2*L3*ddq2*m5*cos(q3) + \\
& L3*L4*ddq2*m4*cos(q4) + 2*L3*L4*ddq2*m5*cos(q4) + L3*L4*ddq3*m4*cos(q4) + \\
& 2*L3*L4*ddq3*m5*cos(q4) + (L3*L4*ddq4*m4*cos(q4))/2 + L3*L4*ddq4*m5*cos(q4) + \\
& (3*L2*L5*dq1^2*m5*cos(2*q2 + q3 + q4))/4 + 3*L3*L5*ddq2*m5*sin(q4) + \\
& 3*L3*L5*ddq3*m5*sin(q4) + (3*L3*L5*ddq4*m5*sin(q4))/2 - (L2*L4*dq1^2*m4*sin(2*q2 \\
& + q3 + q4))/4 - (L2*L4*dq1^2*m5*sin(2*q2 + q3 + q4))/2 + (L3*L6*dq1^2*m3*cos(q2 + \\
& q3))/2 + L3*L6*dq1^2*m4*cos(q2 + q3) - (3*L2*L5*dq1^2*m5*cos(q3 + q4))/4 + \\
& L3*L6*dq1^2*m5*cos(q2 + q3) - (3*L2*L5*dq2^2*m5*cos(q3 + q4))/2 + \\
& (L2*L4*dq1^2*m4*sin(q3 + q4))/4 + (L2*L4*dq1^2*m5*sin(q3 + q4))/2 + \\
& (L2*L4*dq2^2*m4*sin(q3 + q4))/2 + L2*L4*dq2^2*m5*sin(q3 + q4) + \\
& (3*L3*L5*dq4^2*m5*cos(q4))/2 + (3*L3*L5*dq1^2*m5*cos(2*q2 + 2*q3 + q4))/2 + \\
& 3*L3*L5*dq2*dq4*m5*cos(q4) + 3*L3*L5*dq3*dq4*m5*cos(q4) - \\
& L3*L4*dq2*dq4*m4*sin(q4) - 2*L3*L4*dq2*dq4*m5*sin(q4) - \\
& L3*L4*dq3*dq4*m4*sin(q4) - 2*L3*L4*dq3*dq4*m5*sin(q4)
\end{aligned}$$

tau 2 =

$$\begin{aligned}
& I2*ddq2 + I3*ddq2 + I3*ddq3 + I4*ddq2 + I4*ddq3 + I5*ddq2 + I4*ddq4 + I5*ddq3 + I5*ddq4 \\
& + Iext*ddq2 + Iext*ddq3 + Iext*ddq4 + b2*dq2 + (L2^2*ddq2*m2)/4 + L2^2*ddq2*m3 + \\
& L2^2*ddq2*m4 + (L3^2*ddq2*m3)/4 + L2^2*ddq2*m5 + L3^2*ddq2*m4 + \\
& (L3^2*ddq3*m3)/4 + L3^2*ddq2*m5 + L3^2*ddq3*m4 + (L4^2*ddq2*m4)/4 + \\
& L3^2*ddq3*m5 + L4^2*ddq2*m5 + (L4^2*ddq3*m4)/4 + L4^2*ddq3*m5 + \\
& (L4^2*ddq4*m4)/4 + (9*L5^2*ddq2*m5)/4 + L4^2*ddq4*m5 + (9*L5^2*ddq3*m5)/4 + \\
& (9*L5^2*ddq4*m5)/4 + (L4*g*m4*cos(q1 + q2 + q3 + q4))/4 + (L4*g*m5*cos(q1 + q2 + q3 \\
& + q4))/2 + (3*L5*g*m5*sin(q1 + q2 + q3 + q4))/4 + (L2*g*m2*cos(q1 + q2))/4 + \\
& (L2*g*m3*cos(q1 + q2))/2 + (L2*g*m4*cos(q1 + q2))/2 + (L2*g*m5*cos(q1 + q2))/2 + \\
& (L4*g*m4*cos(q1 - q2 - q3 - q4))/4 + (L4*g*m5*cos(q1 - q2 - q3 - q4))/2 - \\
& (3*L5*g*m5*sin(q1 - q2 - q3 - q4))/4 + (L3*g*m3*cos(q1 - q2 - q3))/4 + (L3*g*m4*cos(q1 - \\
& q2 - q3))/2 + (L3*g*m5*cos(q1 - q2 - q3))/2 - (L2^2*dq1^2*m2*sin(2*q2))/8 - \\
& (L2^2*dq1^2*m3*sin(2*q2))/2 - (L2^2*dq1^2*m4*sin(2*q2))/2 - \\
& (L2^2*dq1^2*m5*sin(2*q2))/2 - (L4^2*dq1^2*m4*sin(2*q2 + 2*q3 + 2*q4))/8 - \\
& (L4^2*dq1^2*m5*sin(2*q2 + 2*q3 + 2*q4))/2 + (9*L5^2*dq1^2*m5*sin(2*q2 + 2*q3 + \\
& 2*q4))/8 + (L2*g*m2*cos(q1 - q2))/4 + (L2*g*m3*cos(q1 - q2))/2 + (L2*g*m4*cos(q1 - \\
& q2))/2 + (L2*g*m5*cos(q1 - q2))/2 - I5*dq1*dq5*cos(q2 + q3 + q4) - Iext*dq1*dq5*cos(q2 \\
& + q3 + q4) + (L3*g*m3*cos(q1 + q2 + q3))/4 + (L3*g*m4*cos(q1 + q2 + q3))/2 + \\
& (L3*g*m5*cos(q1 + q2 + q3))/2 - (L3^2*dq1^2*m3*sin(2*q2 + 2*q3))/8 -
\end{aligned}$$

$$\begin{aligned}
& (L3^2 \cdot dq1^2 \cdot m4 \cdot \sin(2 \cdot q2 + 2 \cdot q3))/2 - (L3^2 \cdot dq1^2 \cdot m5 \cdot \sin(2 \cdot q2 + 2 \cdot q3))/2 - \\
& (L2 \cdot L3 \cdot dq3^2 \cdot m3 \cdot \sin(q3))/2 - L2 \cdot L3 \cdot dq3^2 \cdot m4 \cdot \sin(q3) - L2 \cdot L3 \cdot dq3^2 \cdot m5 \cdot \sin(q3) - \\
& (L3 \cdot L4 \cdot dq4^2 \cdot m4 \cdot \sin(q4))/2 - L3 \cdot L4 \cdot dq4^2 \cdot m5 \cdot \sin(q4) - (L3 \cdot L4 \cdot dq1^2 \cdot m4 \cdot \sin(2 \cdot q2 + \\
& 2 \cdot q3 + q4))/2 - L3 \cdot L4 \cdot dq1^2 \cdot m5 \cdot \sin(2 \cdot q2 + 2 \cdot q3 + q4) - (L2 \cdot L3 \cdot dq1^2 \cdot m3 \cdot \sin(2 \cdot q2 + \\
& q3))/2 - L2 \cdot L3 \cdot dq1^2 \cdot m4 \cdot \sin(2 \cdot q2 + q3) - L2 \cdot L3 \cdot dq1^2 \cdot m5 \cdot \sin(2 \cdot q2 + q3) + \\
& (L4 \cdot L6 \cdot dq1^2 \cdot m4 \cdot \cos(q2 + q3 + q4))/2 + L4 \cdot L6 \cdot dq1^2 \cdot m5 \cdot \cos(q2 + q3 + q4) + \\
& (3 \cdot L5 \cdot L6 \cdot dq1^2 \cdot m5 \cdot \sin(q2 + q3 + q4))/2 + (3 \cdot L4 \cdot L5 \cdot dq1^2 \cdot m5 \cdot \cos(2 \cdot q2 + 2 \cdot q3 + \\
& 2 \cdot q4))/2 + L2 \cdot L4 \cdot ddq2 \cdot m4 \cdot \cos(q3 + q4) + 2 \cdot L2 \cdot L4 \cdot ddq2 \cdot m5 \cdot \cos(q3 + q4) + \\
& (L2 \cdot L4 \cdot ddq3 \cdot m4 \cdot \cos(q3 + q4))/2 + L2 \cdot L4 \cdot ddq3 \cdot m5 \cdot \cos(q3 + q4) + \\
& (L2 \cdot L4 \cdot ddq4 \cdot m4 \cdot \cos(q3 + q4))/2 + L2 \cdot L4 \cdot ddq4 \cdot m5 \cdot \cos(q3 + q4) + \\
& 3 \cdot L2 \cdot L5 \cdot ddq2 \cdot m5 \cdot \sin(q3 + q4) + (3 \cdot L2 \cdot L5 \cdot ddq3 \cdot m5 \cdot \sin(q3 + q4))/2 + \\
& (3 \cdot L2 \cdot L5 \cdot ddq4 \cdot m5 \cdot \sin(q3 + q4))/2 + L2 \cdot L3 \cdot ddq2 \cdot m3 \cdot \cos(q3) + \\
& 2 \cdot L2 \cdot L3 \cdot ddq2 \cdot m4 \cdot \cos(q3) + (L2 \cdot L3 \cdot ddq3 \cdot m3 \cdot \cos(q3))/2 + 2 \cdot L2 \cdot L3 \cdot ddq2 \cdot m5 \cdot \cos(q3) + \\
& L2 \cdot L3 \cdot ddq3 \cdot m4 \cdot \cos(q3) + L2 \cdot L3 \cdot ddq3 \cdot m5 \cdot \cos(q3) + L3 \cdot L4 \cdot ddq2 \cdot m4 \cdot \cos(q4) + \\
& 2 \cdot L3 \cdot L4 \cdot ddq2 \cdot m5 \cdot \cos(q4) + L3 \cdot L4 \cdot ddq3 \cdot m4 \cdot \cos(q4) + 2 \cdot L3 \cdot L4 \cdot ddq3 \cdot m5 \cdot \cos(q4) + \\
& (L3 \cdot L4 \cdot ddq4 \cdot m4 \cdot \cos(q4))/2 + L3 \cdot L4 \cdot ddq4 \cdot m5 \cdot \cos(q4) + (3 \cdot L2 \cdot L5 \cdot dq1^2 \cdot m5 \cdot \cos(2 \cdot q2 \\
& + q3 + q4))/2 + 3 \cdot L3 \cdot L5 \cdot ddq2 \cdot m5 \cdot \sin(q4) + 3 \cdot L3 \cdot L5 \cdot ddq3 \cdot m5 \cdot \sin(q4) + \\
& (3 \cdot L3 \cdot L5 \cdot ddq4 \cdot m5 \cdot \sin(q4))/2 - (L2 \cdot L4 \cdot dq1^2 \cdot m4 \cdot \sin(2 \cdot q2 + q3 + q4))/2 - \\
& L2 \cdot L4 \cdot dq1^2 \cdot m5 \cdot \sin(2 \cdot q2 + q3 + q4) + (L3 \cdot L6 \cdot dq1^2 \cdot m3 \cdot \cos(q2 + q3))/2 + \\
& L3 \cdot L6 \cdot dq1^2 \cdot m4 \cdot \cos(q2 + q3) + L3 \cdot L6 \cdot dq1^2 \cdot m5 \cdot \cos(q2 + q3) + \\
& (3 \cdot L2 \cdot L5 \cdot dq3^2 \cdot m5 \cdot \cos(q3 + q4))/2 + (3 \cdot L2 \cdot L5 \cdot dq4^2 \cdot m5 \cdot \cos(q3 + q4))/2 - \\
& (L2 \cdot L4 \cdot dq3^2 \cdot m4 \cdot \sin(q3 + q4))/2 - L2 \cdot L4 \cdot dq3^2 \cdot m5 \cdot \sin(q3 + q4) - \\
& (L2 \cdot L4 \cdot dq4^2 \cdot m4 \cdot \sin(q3 + q4))/2 - L2 \cdot L4 \cdot dq4^2 \cdot m5 \cdot \sin(q3 + q4) + \\
& (L2 \cdot L6 \cdot dq1^2 \cdot m2 \cdot \cos(q2))/2 + L2 \cdot L6 \cdot dq1^2 \cdot m3 \cdot \cos(q2) + L2 \cdot L6 \cdot dq1^2 \cdot m4 \cdot \cos(q2) + \\
& L2 \cdot L6 \cdot dq1^2 \cdot m5 \cdot \cos(q2) + (3 \cdot L3 \cdot L5 \cdot dq4^2 \cdot m5 \cdot \cos(q4))/2 + \\
& (3 \cdot L3 \cdot L5 \cdot dq1^2 \cdot m5 \cdot \cos(2 \cdot q2 + 2 \cdot q3 + q4))/2 + 3 \cdot L3 \cdot L5 \cdot dq2 \cdot dq4 \cdot m5 \cdot \cos(q4) + \\
& 3 \cdot L3 \cdot L5 \cdot dq3 \cdot dq4 \cdot m5 \cdot \cos(q4) - L2 \cdot L3 \cdot dq2 \cdot dq3 \cdot m3 \cdot \sin(q3) - \\
& 2 \cdot L2 \cdot L3 \cdot dq2 \cdot dq3 \cdot m4 \cdot \sin(q3) - 2 \cdot L2 \cdot L3 \cdot dq2 \cdot dq3 \cdot m5 \cdot \sin(q3) - \\
& L3 \cdot L4 \cdot dq2 \cdot dq4 \cdot m4 \cdot \sin(q4) - 2 \cdot L3 \cdot L4 \cdot dq2 \cdot dq4 \cdot m5 \cdot \sin(q4) - \\
& L3 \cdot L4 \cdot dq3 \cdot dq4 \cdot m4 \cdot \sin(q4) - 2 \cdot L3 \cdot L4 \cdot dq3 \cdot dq4 \cdot m5 \cdot \sin(q4) + \\
& 3 \cdot L2 \cdot L5 \cdot dq2 \cdot dq3 \cdot m5 \cdot \cos(q3 + q4) + 3 \cdot L2 \cdot L5 \cdot dq2 \cdot dq4 \cdot m5 \cdot \cos(q3 + q4) + \\
& 3 \cdot L2 \cdot L5 \cdot dq3 \cdot dq4 \cdot m5 \cdot \cos(q3 + q4) - L2 \cdot L4 \cdot dq2 \cdot dq3 \cdot m4 \cdot \sin(q3 + q4) - \\
& 2 \cdot L2 \cdot L4 \cdot dq2 \cdot dq3 \cdot m5 \cdot \sin(q3 + q4) - L2 \cdot L4 \cdot dq2 \cdot dq4 \cdot m4 \cdot \sin(q3 + q4) - \\
& 2 \cdot L2 \cdot L4 \cdot dq2 \cdot dq4 \cdot m5 \cdot \sin(q3 + q4) - L2 \cdot L4 \cdot dq3 \cdot dq4 \cdot m4 \cdot \sin(q3 + q4) - \\
& 2 \cdot L2 \cdot L4 \cdot dq3 \cdot dq4 \cdot m5 \cdot \sin(q3 + q4)
\end{aligned}$$

tau 1 =

$$\begin{aligned}
 & I1*ddq1 + I2*ddq1 + I3*ddq1 + I4*ddq1 + I5*ddq1 + I_{ext}*ddq1 + b1*dq1 + I5*ddq5*\sin(q2 \\
 & + q3 + q4) + I_{ext}*ddq5*\sin(q2 + q3 + q4) + (L2^2*ddq1*m2)/8 + (L2^2*ddq1*m3)/2 + \\
 & (L2^2*ddq1*m4)/2 + (L3^2*ddq1*m3)/8 + (L2^2*ddq1*m5)/2 + (L3^2*ddq1*m4)/2 + \\
 & L6^2*ddq1*m1 + (L3^2*ddq1*m5)/2 + (L4^2*ddq1*m4)/8 + L6^2*ddq1*m2 + \\
 & (L4^2*ddq1*m5)/2 + L6^2*ddq1*m3 + (9*L5^2*ddq1*m5)/8 + L6^2*ddq1*m4 + \\
 & L6^2*ddq1*m5 + (L4*g*m4*\cos(q1 + q2 + q3 + q4))/4 + (L4*g*m5*\cos(q1 + q2 + q3 + q4))/2 \\
 & - (L2^2*ddq1*m2*\cos(2*q2))/8 - (L2^2*ddq1*m3*\cos(2*q2))/2 - \\
 & (L2^2*ddq1*m4*\cos(2*q2))/2 - (L2^2*ddq1*m5*\cos(2*q2))/2 - (L4^2*ddq1*m4*\cos(2*q2 \\
 & + 2*q3 + 2*q4))/8 - (L4^2*ddq1*m5*\cos(2*q2 + 2*q3 + 2*q4))/2 + \\
 & (9*L5^2*ddq1*m5*\cos(2*q2 + 2*q3 + 2*q4))/8 + (3*L5*g*m5*\sin(q1 + q2 + q3 + q4))/4 + \\
 & (L2*g*m2*\cos(q1 + q2))/4 + (L2*g*m3*\cos(q1 + q2))/2 + (L2*g*m4*\cos(q1 + q2))/2 + \\
 & (L2*g*m5*\cos(q1 + q2))/2 - (L4*g*m4*\cos(q1 - q2 - q3 - q4))/4 - (L4*g*m5*\cos(q1 - q2 - q3 \\
 & - q4))/2 + (3*L5*g*m5*\sin(q1 - q2 - q3 - q4))/4 - (L3^2*ddq1*m3*\cos(2*q2 + 2*q3))/8 - \\
 & (L3^2*ddq1*m4*\cos(2*q2 + 2*q3))/2 - (L3^2*ddq1*m5*\cos(2*q2 + 2*q3))/2 - \\
 & (L3*g*m3*\cos(q1 - q2 - q3))/4 - (L3*g*m4*\cos(q1 - q2 - q3))/2 - (L3*g*m5*\cos(q1 - q2 - \\
 & q3))/2 + L6*g*m1*\sin(q1) + L6*g*m2*\sin(q1) + L6*g*m3*\sin(q1) + L6*g*m4*\sin(q1) + \\
 & L6*g*m5*\sin(q1) - (L2*g*m2*\cos(q1 - q2))/4 - (L2*g*m3*\cos(q1 - q2))/2 - (L2*g*m4*\cos(q1 \\
 & - q2))/2 - (L2*g*m5*\cos(q1 - q2))/2 + I5*dq2*dq5*\cos(q2 + q3 + q4) + I5*dq3*dq5*\cos(q2 \\
 & + q3 + q4) + I5*dq4*dq5*\cos(q2 + q3 + q4) + I_{ext}*dq2*dq5*\cos(q2 + q3 + q4) + \\
 & I_{ext}*dq3*dq5*\cos(q2 + q3 + q4) + I_{ext}*dq4*dq5*\cos(q2 + q3 + q4) + (L3*g*m3*\cos(q1 + q2 \\
 & + q3))/4 + (L3*g*m4*\cos(q1 + q2 + q3))/2 + (L3*g*m5*\cos(q1 + q2 + q3))/2 - \\
 & (3*L4*L5*ddq1*m5*\sin(2*q2 + 2*q3 + 2*q4))/2 - (L2*L4*ddq1*m4*\cos(2*q2 + q3 + q4))/2 \\
 & - L2*L4*ddq1*m5*\cos(2*q2 + q3 + q4) - (3*L2*L5*ddq1*m5*\sin(2*q2 + q3 + q4))/2 + \\
 & (L2*L4*ddq1*m4*\cos(q3 + q4))/2 + L2*L4*ddq1*m5*\cos(q3 + q4) - \\
 & L3*L6*ddq1*m3*\sin(q2 + q3) - 2*L3*L6*ddq1*m4*\sin(q2 + q3) + \\
 & (3*L2*L5*ddq1*m5*\sin(q3 + q4))/2 - 2*L3*L6*ddq1*m5*\sin(q2 + q3) + \\
 & (L2^2*dq1*dq2*m2*\sin(2*q2))/4 + L2^2*dq1*dq2*m3*\sin(2*q2) + \\
 & L2^2*dq1*dq2*m4*\sin(2*q2) + L2^2*dq1*dq2*m5*\sin(2*q2) + \\
 & (L4^2*dq1*dq2*m4*\sin(2*q2 + 2*q3 + 2*q4))/4 + L4^2*dq1*dq2*m5*\sin(2*q2 + 2*q3 + \\
 & 2*q4) + (L4^2*dq1*dq3*m4*\sin(2*q2 + 2*q3 + 2*q4))/4 + L4^2*dq1*dq3*m5*\sin(2*q2 + \\
 & 2*q3 + 2*q4) + (L4^2*dq1*dq4*m4*\sin(2*q2 + 2*q3 + 2*q4))/4 - \\
 & (9*L5^2*dq1*dq2*m5*\sin(2*q2 + 2*q3 + 2*q4))/4 + L4^2*dq1*dq4*m5*\sin(2*q2 + 2*q3 \\
 & + 2*q4) - (9*L5^2*dq1*dq3*m5*\sin(2*q2 + 2*q3 + 2*q4))/4 - \\
 & (9*L5^2*dq1*dq4*m5*\sin(2*q2 + 2*q3 + 2*q4))/4 + (L2*L3*ddq1*m3*\cos(q3))/2 + \\
 & L2*L3*ddq1*m4*\cos(q3) + L2*L3*ddq1*m5*\cos(q3) + (L3*L4*ddq1*m4*\cos(q4))/2 + \\
 & L3*L4*ddq1*m5*\cos(q4) - (L3*L4*ddq1*m4*\cos(2*q2 + 2*q3 + q4))/2 - \\
 & L3*L4*ddq1*m5*\cos(2*q2 + 2*q3 + q4) - L2*L6*ddq1*m2*\sin(q2) -
 \end{aligned}$$

$$\begin{aligned}
& 2*L2*L6*ddq1*m3*\sin(q2) - 2*L2*L6*ddq1*m4*\sin(q2) - 2*L2*L6*ddq1*m5*\sin(q2) + \\
& (3*L3*L5*ddq1*m5*\sin(q4))/2 - (3*L3*L5*ddq1*m5*\sin(2*q2 + 2*q3 + q4))/2 + \\
& (L3^2*dq1*dq2*m3*\sin(2*q2 + 2*q3))/4 + L3^2*dq1*dq2*m4*\sin(2*q2 + 2*q3) + \\
& (L3^2*dq1*dq3*m3*\sin(2*q2 + 2*q3))/4 + L3^2*dq1*dq2*m5*\sin(2*q2 + 2*q3) + \\
& L3^2*dq1*dq3*m4*\sin(2*q2 + 2*q3) + L3^2*dq1*dq3*m5*\sin(2*q2 + 2*q3) - \\
& (L2*L3*ddq1*m3*\cos(2*q2 + q3))/2 - L2*L3*ddq1*m4*\cos(2*q2 + q3) - \\
& L2*L3*ddq1*m5*\cos(2*q2 + q3) + 3*L5*L6*ddq1*m5*\cos(q2 + q3 + q4) - \\
& L4*L6*ddq1*m4*\sin(q2 + q3 + q4) - 2*L4*L6*ddq1*m5*\sin(q2 + q3 + q4) - \\
& L2*L6*dq1*dq2*m2*\cos(q2) - 2*L2*L6*dq1*dq2*m3*\cos(q2) - \\
& 2*L2*L6*dq1*dq2*m4*\cos(q2) - 2*L2*L6*dq1*dq2*m5*\cos(q2) + \\
& (3*L3*L5*dq1*dq4*m5*\cos(q4))/2 - 3*L3*L5*dq1*dq2*m5*\cos(2*q2 + 2*q3 + q4) - \\
& 3*L3*L5*dq1*dq3*m5*\cos(2*q2 + 2*q3 + q4) - (3*L3*L5*dq1*dq4*m5*\cos(2*q2 + 2*q3 + \\
& q4))/2 - (L2*L3*dq1*dq3*m3*\sin(q3))/2 - L2*L3*dq1*dq3*m4*\sin(q3) - \\
& L2*L3*dq1*dq3*m5*\sin(q3) - (L3*L4*dq1*dq4*m4*\sin(q4))/2 - \\
& L3*L4*dq1*dq4*m5*\sin(q4) + L3*L4*dq1*dq2*m4*\sin(2*q2 + 2*q3 + q4) + \\
& 2*L3*L4*dq1*dq2*m5*\sin(2*q2 + 2*q3 + q4) + L3*L4*dq1*dq3*m4*\sin(2*q2 + 2*q3 + q4) \\
& + 2*L3*L4*dq1*dq3*m5*\sin(2*q2 + 2*q3 + q4) + (L3*L4*dq1*dq4*m4*\sin(2*q2 + 2*q3 + \\
& q4))/2 + L3*L4*dq1*dq4*m5*\sin(2*q2 + 2*q3 + q4) + L2*L3*dq1*dq2*m3*\sin(2*q2 + q3) \\
& + 2*L2*L3*dq1*dq2*m4*\sin(2*q2 + q3) + (L2*L3*dq1*dq3*m3*\sin(2*q2 + q3))/2 + \\
& 2*L2*L3*dq1*dq2*m5*\sin(2*q2 + q3) + L2*L3*dq1*dq3*m4*\sin(2*q2 + q3) + \\
& L2*L3*dq1*dq3*m5*\sin(2*q2 + q3) - L4*L6*dq1*dq2*m4*\cos(q2 + q3 + q4) - \\
& 2*L4*L6*dq1*dq2*m5*\cos(q2 + q3 + q4) - L4*L6*dq1*dq3*m4*\cos(q2 + q3 + q4) - \\
& 2*L4*L6*dq1*dq3*m5*\cos(q2 + q3 + q4) - L4*L6*dq1*dq4*m4*\cos(q2 + q3 + q4) - \\
& 2*L4*L6*dq1*dq4*m5*\cos(q2 + q3 + q4) - 3*L5*L6*dq1*dq2*m5*\sin(q2 + q3 + q4) - \\
& 3*L5*L6*dq1*dq3*m5*\sin(q2 + q3 + q4) - 3*L5*L6*dq1*dq4*m5*\sin(q2 + q3 + q4) - \\
& 3*L4*L5*dq1*dq2*m5*\cos(2*q2 + 2*q3 + 2*q4) - 3*L4*L5*dq1*dq3*m5*\cos(2*q2 + 2*q3 \\
& + 2*q4) - 3*L4*L5*dq1*dq4*m5*\cos(2*q2 + 2*q3 + 2*q4) - \\
& 3*L2*L5*dq1*dq2*m5*\cos(2*q2 + q3 + q4) - (3*L2*L5*dq1*dq3*m5*\cos(2*q2 + q3 + \\
& q4))/2 - (3*L2*L5*dq1*dq4*m5*\cos(2*q2 + q3 + q4))/2 + L2*L4*dq1*dq2*m4*\sin(2*q2 + \\
& q3 + q4) + 2*L2*L4*dq1*dq2*m5*\sin(2*q2 + q3 + q4) + (L2*L4*dq1*dq3*m4*\sin(2*q2 + \\
& q3 + q4))/2 + L2*L4*dq1*dq3*m5*\sin(2*q2 + q3 + q4) + (L2*L4*dq1*dq4*m4*\sin(2*q2 + \\
& q3 + q4))/2 + L2*L4*dq1*dq4*m5*\sin(2*q2 + q3 + q4) - L3*L6*dq1*dq2*m3*\cos(q2 + q3) - \\
& 2*L3*L6*dq1*dq2*m4*\cos(q2 + q3) - L3*L6*dq1*dq3*m3*\cos(q2 + q3) - \\
& 2*L3*L6*dq1*dq2*m5*\cos(q2 + q3) - 2*L3*L6*dq1*dq3*m4*\cos(q2 + q3) + \\
& (3*L2*L5*dq1*dq3*m5*\cos(q3 + q4))/2 - 2*L3*L6*dq1*dq3*m5*\cos(q2 + q3) + \\
& (3*L2*L5*dq1*dq4*m5*\cos(q3 + q4))/2 - (L2*L4*dq1*dq3*m4*\sin(q3 + q4))/2 - \\
& L2*L4*dq1*dq3*m5*\sin(q3 + q4) - (L2*L4*dq1*dq4*m4*\sin(q3 + q4))/2 - \\
& L2*L4*dq1*dq4*m5*\sin(q3 + q4)
\end{aligned}$$

3.4 PLANIFICACION DE TRAYECTORIAS.

Para realizar esta animación se necesitó realizar una planificación simple de la trayectoria que debía seguir el robot. En aquel caso no se tuvo en consideración ninguna restricción sobre el camino que debía trazar el robot, y simplemente se obtuvieron una serie de posiciones por las que fue pasando el robot cuando se resolvía su cinemática inversa. En ningún momento se tuvo en cuenta la realidad física de los actuadores que proporcionan el movimiento a los eslabones, y sus limitaciones de proporcionar velocidades instantáneas con aceleraciones infinitas.

Asimismo en aquel ejemplo tan sólo se consideró la planificación en posición, sin observar los cambios en la orientación de la herramienta. La realidad del problema de planificación de trayectorias exige sin embargo tener en consideración las prestaciones reales de los actuadores, de tal manera que el movimiento del robot sea un movimiento suave y coordinado. Para llegar a obtener un planificador que funcione correctamente, los pasos a seguir son los siguientes:

1. Estudiar las necesidades de movimiento especificadas por el usuario o por los sensores propios del sistema robotizado, evitando colisiones con el entorno etc., obteniendo una expresión analítica en coordenadas cartesianas de la trayectoria deseada en función del tiempo (libre de colisiones).
2. Muestrear la trayectoria anterior en una serie finita de *puntos nudo* de control que se utilizan como puntos inicial y final de cada segmento. Cada uno de estos puntos está especificado por sus componentes cartesianas de posición y orientación $(x,y,z,\alpha,\beta,\gamma)$.
3. Pasar cada uno de estos puntos a coordenadas articulares del robot, utilizando para ello la transformación homogénea inversa.

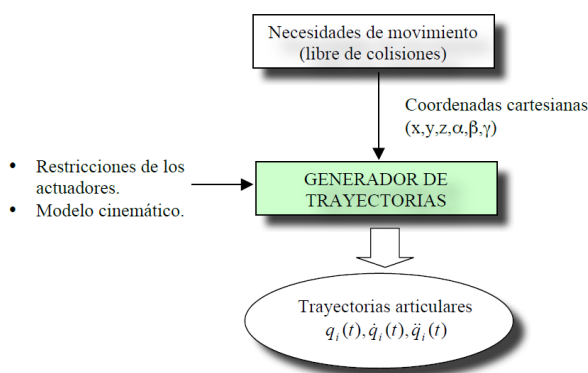


Figura 3.4 Generación de trayectorias

5.2.1.- Espacio cartesiano y Espacio articular

La siguiente figura pretende clarificar al lector el esquema de planificación de trayectorias presentado. En ella se muestra un ejemplo de un robot de 2 gdl. Se quiere que el robot se mueva en línea recta desde la posición cartesiana j1 hasta j4. Para ello se añaden como puntos auxiliares j2

y j_3 en el espacio cartesiano. Cada uno de estos puntos nudo se pasan al espacio articular (en este caso bidimensional). El siguiente paso es realizar la interpolación en el espacio articular, encontrando un polinomio que cumpla con las especificaciones requeridas. La trayectoria cartesiana del robot pasará en este caso por los puntos nudo, si bien entre ellos puede ser que no realice una trayectoria perfectamente recta.

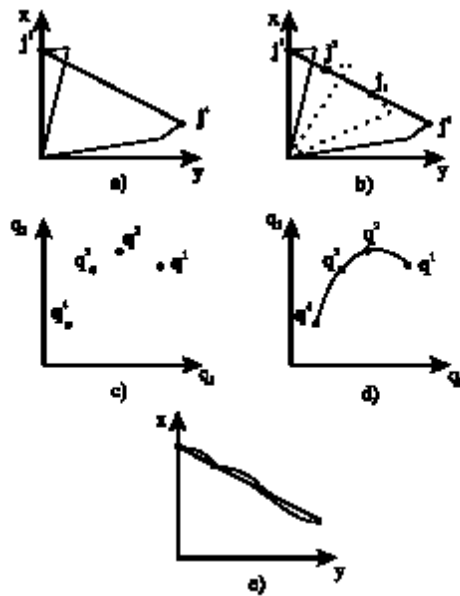


Figura 3.5 Planificación de trayectoria

3.5 DISEÑO SOLIDWORKS BRAZO ROBOT.

Se diseñó el brazo Robot el Solidworks:

En Anexos se presentara el diseño mecánico de cada pieza del brazo Robot.

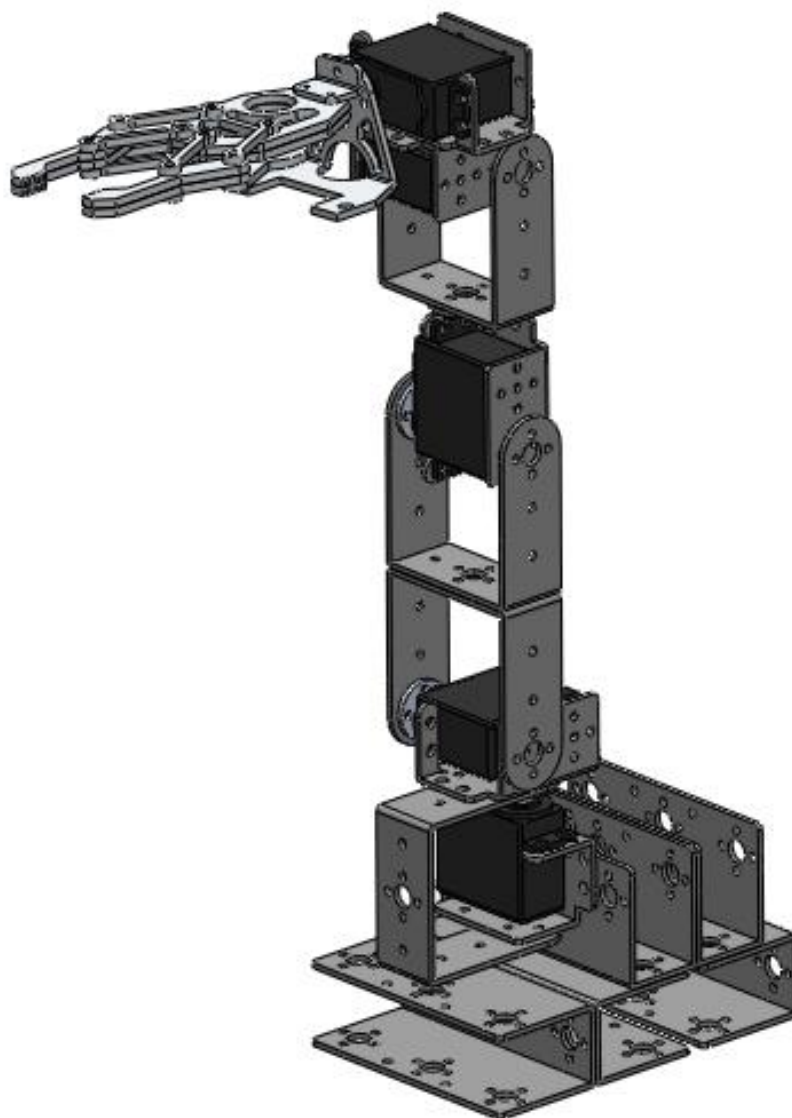


Figura 3.6 Diseño del Brazo Robot en Solidworks Vista Isometrica

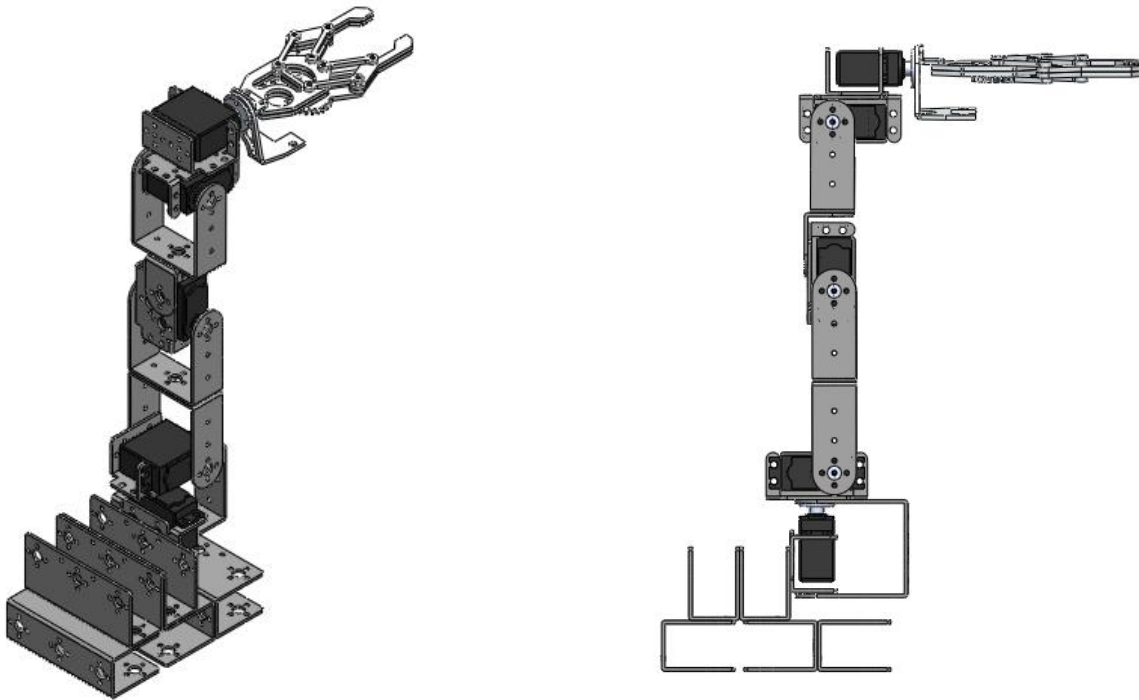


Figura 3.7. Brazo robot 5GDL

3.6 SIMULACION LABVIEW – SOLIDWORKS

Se desarrolló la simulación entre la plataforma de labView y Solidworks del Brazo Robot 5GDL. Primero Se desarrolló el análisis de Movimiento en Solidworks:

Para desarrollar la Simulacion:

- i) Se debe crear un análisis de movimiento cinemático en Solidworks, se activa Solidworks Motion. Y Se configura cada motor rotatorio en cada articulación de movimiento del brazo robot, como se muestra en la Figura 3.7

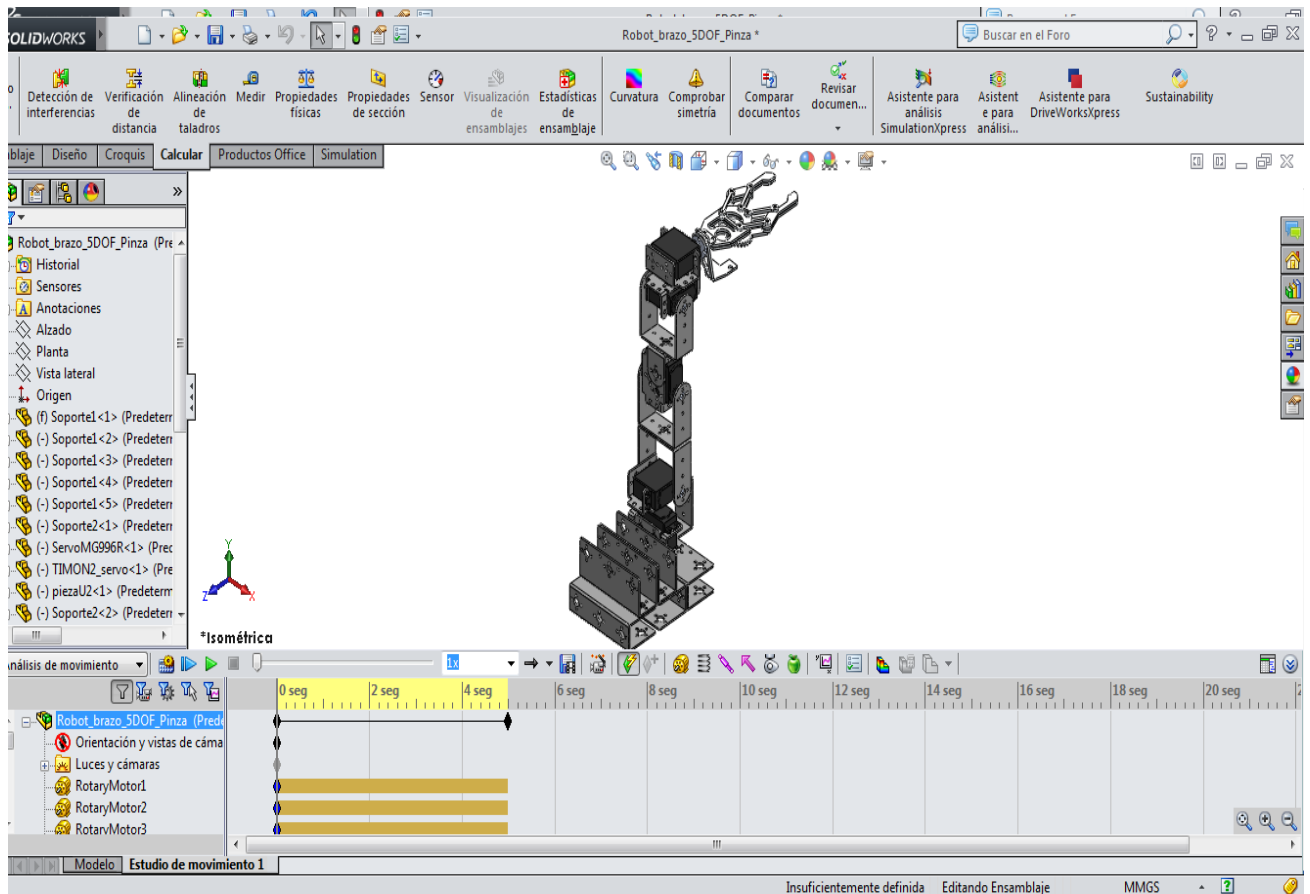


Figura 3.7 Configuración de Análisis de Movimiento EN SOLIDWORKS

- Incluyendo los motores rotatorios de cada articulación.
- ii) Se configura en LabView, primero se Crea un proyecto nuevo, se crea los ejes de Coordenadas, del robot asignándolo a cada motor rotario creado en solidworks, como se muestra en la figura 3.8

Luego se Programa en Labview:

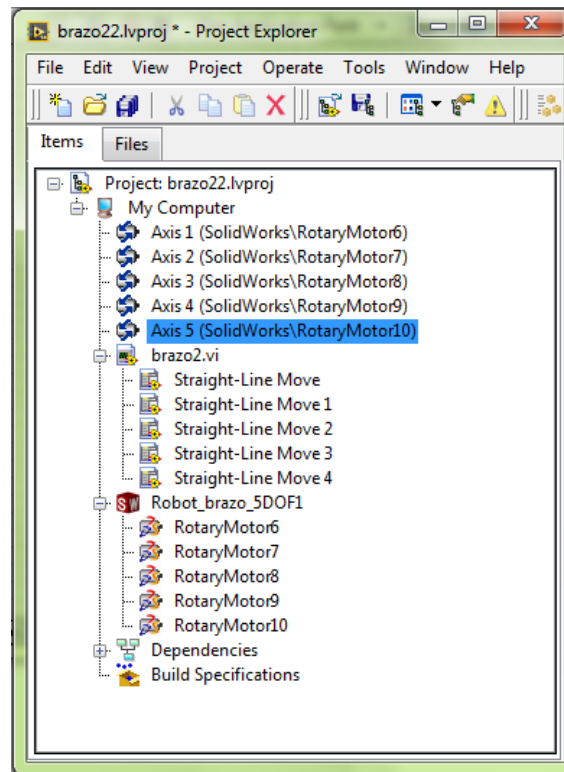


FIGURA 3.8 Configuración en Labview

- iii) Una vez creado el Proyecto Nuevo se desarrolló en Sub VI, la cual se muestra el panel FRONTAL SE TIENE los botones Px, Py, Pz, que te permiten agregar las coordenadas cartesianas de movimiento del robot, Indicador Booleanos (LED) para ver si la información a sido recibida en Solidworks

Se presenta también un botón de estados que te permita agregar 3 estados, (1): estado de reset o posición inicial, (2) estado para cálculo de la cinemática inversa, (3) estado de movimiento del Robot.

Ademas se muestra un Vector q: los Ángulos de las variables Articulares. Un dato importante es que los valores de salida son complejos, la cual se tuvo q convertir a entero, para poder asignarles a los servomortes. Se muestra en la figura 3.9

Se crea un proyecto Nuevo:

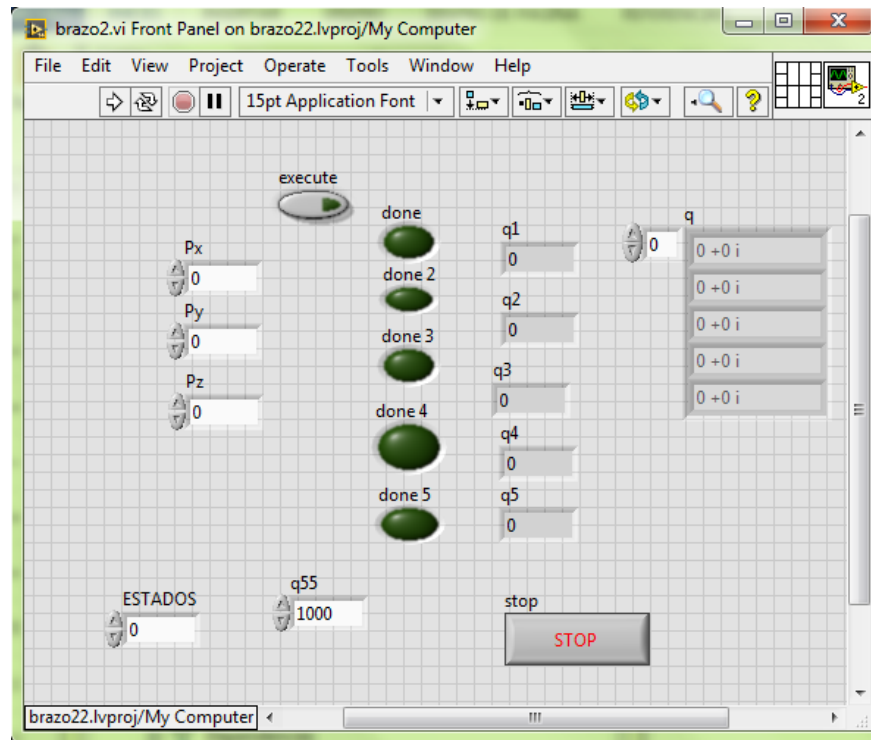


Figura 3.9 Panel Frontal en Labview.

- iv) En el diagrama de bloque de labView. Primero en un bloque while time, que te permite sincronizar el tiempo de ejecución con el de Simulación en solidworks, luego se jala Bloques para configurar cada Servomotor que te permite asignar las posiciones de cada servomotor.

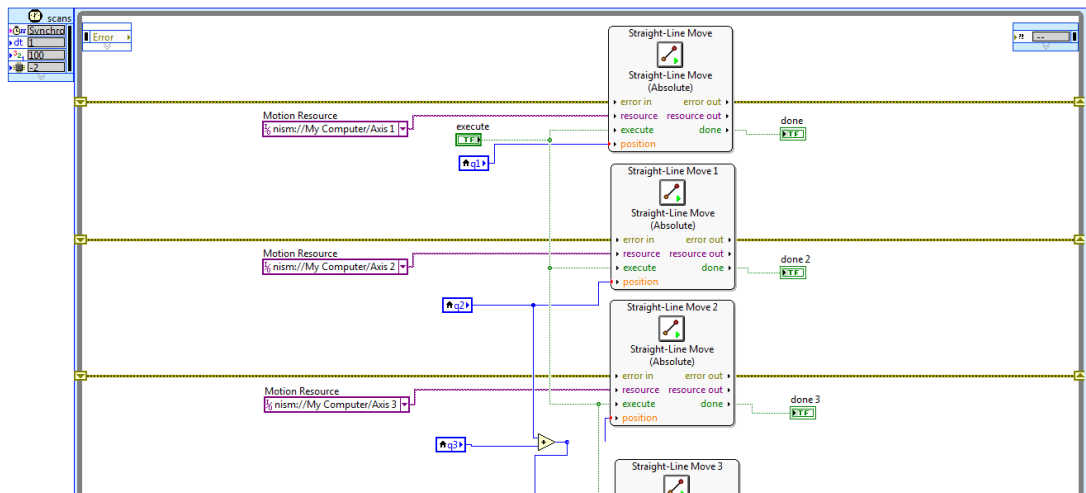


Figura 3.10 Diagrama de Bloques en Labview – Solidwoks.

Los resultados se Muestran en la Figura 3.11 y 3.12

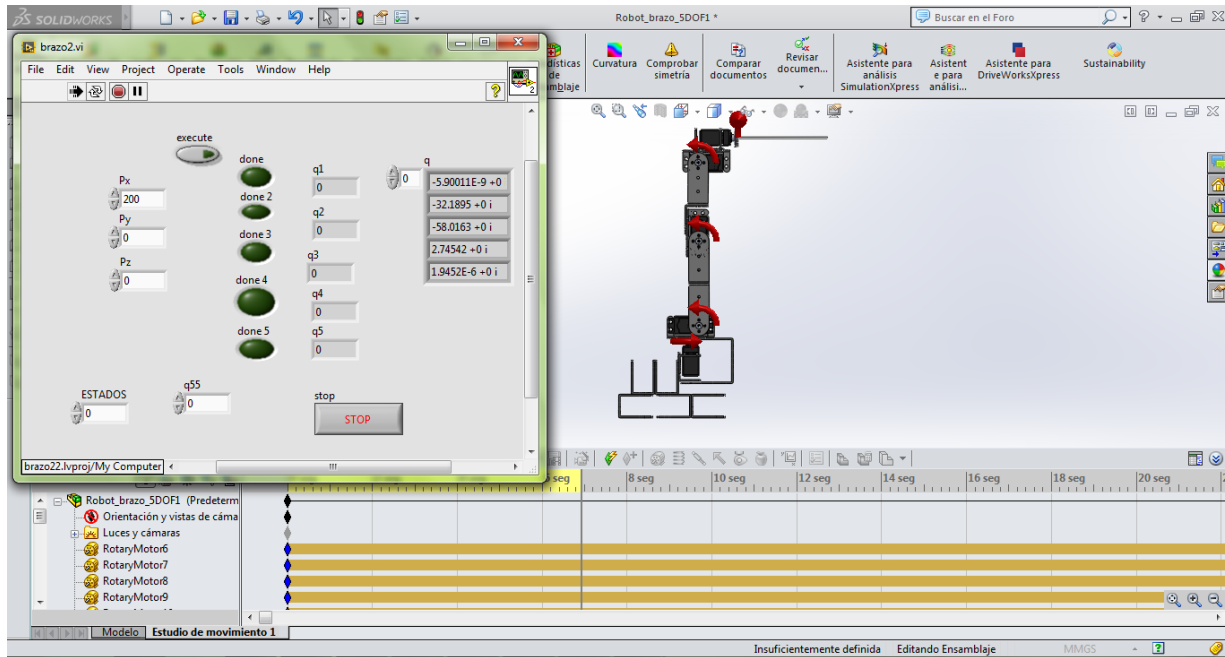


Figura 3.11 Solidworks – Labview Simulación.

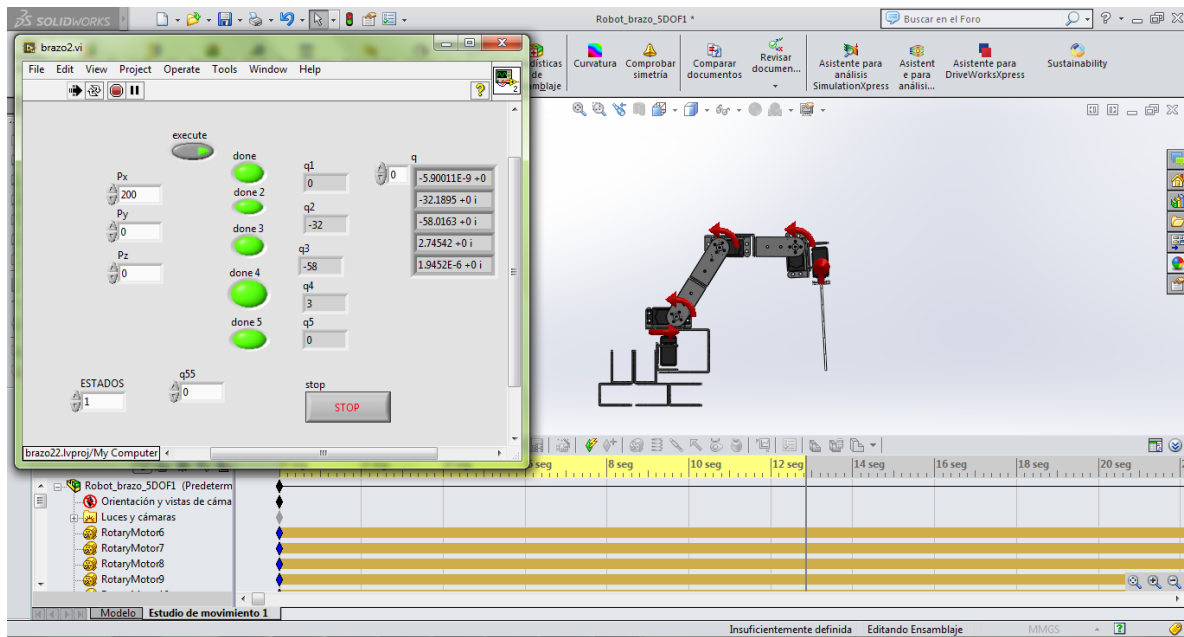


Figura 3.12 SoliWorks – LabView en Movimiento.

Se mostrará un video con la Simulación en movimiento.

3.7 DISEÑO PLATAFORMA LABVIEW – BRAZO ROBOT 5GDL.

La programación se desarrolló en labView para el control del brazo Robot 5GDL.
Se desarrolló un programa de control para el Brazo Robot 5GDL en LabView:

- i) Se quiere calcular la Cinemática inversa. Para esto se utilizar el Modulo de labview – Matlab MathScript que te permite hacer los cálculos correspondientes para evaluar la Cinemática se muestra en la figura 3.13

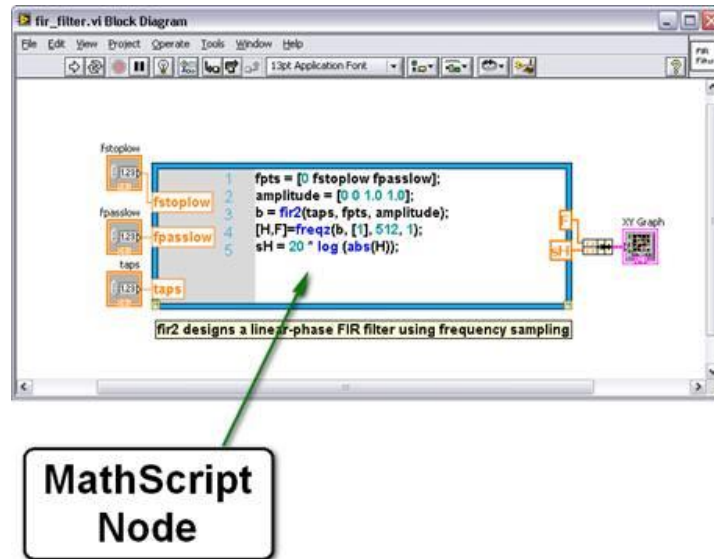


Figura 3.13 Modulo Math Script

- ii) Este desarrollo implica la creación de Subsunciones de Matlab para labView (Math Script)
- iii) El cálculo Dinámico, se desarrolló para la evaluación de los torques y la elección de los Servomotores (no fue incluido en la programación de Labview), se desarrolló en la plataforma de Matlab.
- iv) En la planificación de trayectorias con un control punto a punto, se decidió hacer la programación con una velocidad Constante para articulación en el movimiento, pero a la vez también en la programación se colocó un botón para controlar el tiempo (y así permitiendo controlar la velocidad.) se muestra en la figura 3.14

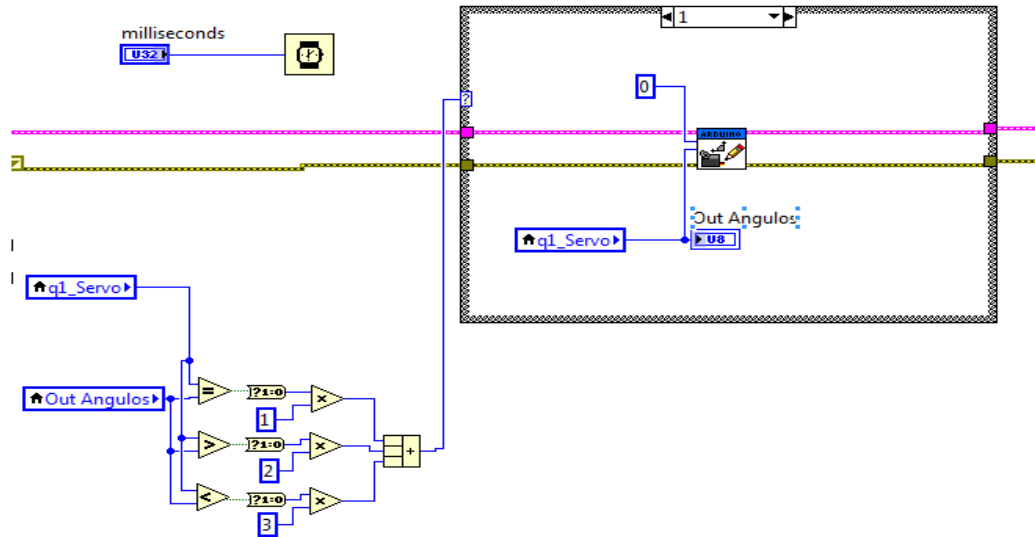


Figura 3.14 bloques para la planificación de trayectorias.

En ese bloque de selector se da 3 opciones :

Opción 1: Permite Asignar la Relación Igual El Set point con la Posicion actual del Servomotor.

Opción 2: Se activa cuando la Relación de Set point mayor a la posición actual , entonces se aumentando la posición actual hasta llegar el set point.

Opción 3: Se activa cuando la Relación de Set point menor a la posición actual , entonces se disminuye la posición actual hasta llegar el set point.

Lo cual se muestra parte del algoritmo en LabView en la figura 3.15

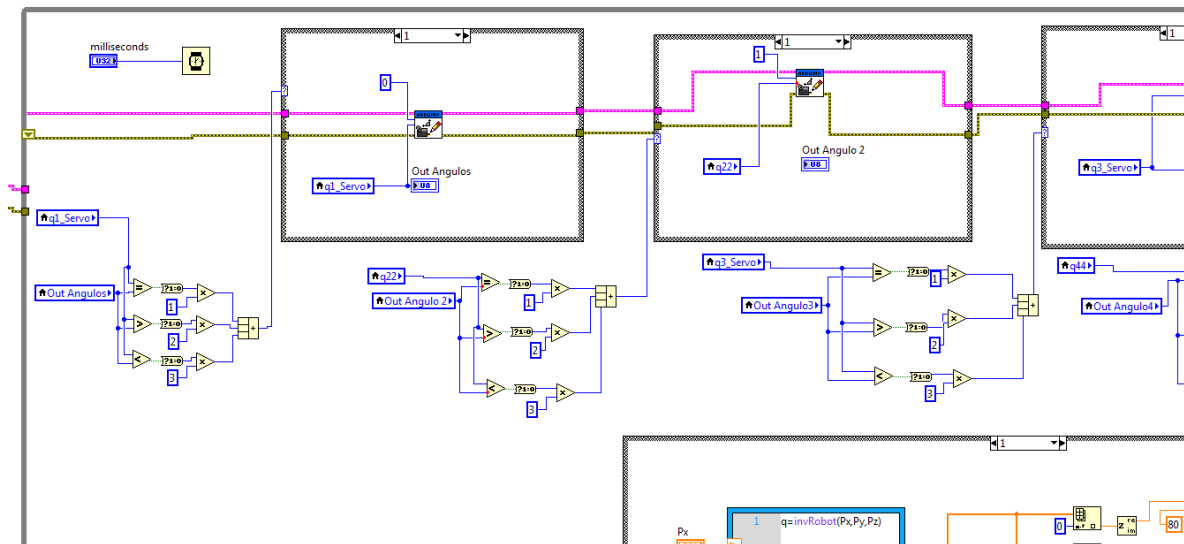


Figura 3.5 Programa de Robot 5GDL - LabView

El panel Frontal Se desarrollo se explicara cada indicador:

- ✓ **VISA resource:** Te permite elegir el puerto COM (Serial) para la comunicación con el Microcontrolador (ARDUINO) – SE UTILIZO EL LIFA DE LABVIEW PARA ARDUINO.
- ✓ La configuración de cada Pin Servo de salida para el control cada Servomotor.
- ✓ Un botón de entrada para configurar el tiempo (en milisegundos) Se recomienda entre 10 – 100 ms.
- ✓ Entrada: Px, Py, Pz : te permite ingresar las posiciones x, y ,z en milímetros el sistema de referencia ubicada en la base del robot (respecto al Sistema de coordenadas establecidas por Denavit hartemberg.)
- ✓ Entrada Estado: te permite variar en 2 opciones 0:1
 - 0: estado inicial del Robot.
 - 1: Estado se mueve a la posición deseada y calculada Px, Py,Pz
 - 3.- Se ejecuta el movimiento del Brazo Robot.
- ✓ Indicador: el Vector de las posiciones “q” donde se muestra q1,q2,q3,q4,q5 la cual se muestra en salida: en formato complejo.
- ✓ Indicador de cada Articulación de los Servomotores: como también los Set Point para cada posición en tiempo REAL.
- ✓ Un botón de entrada para el control de la pinza de la posición 0°-180° calibrada con los t
- ✓ Un Botón de Stop. Para finalizar el proceso.

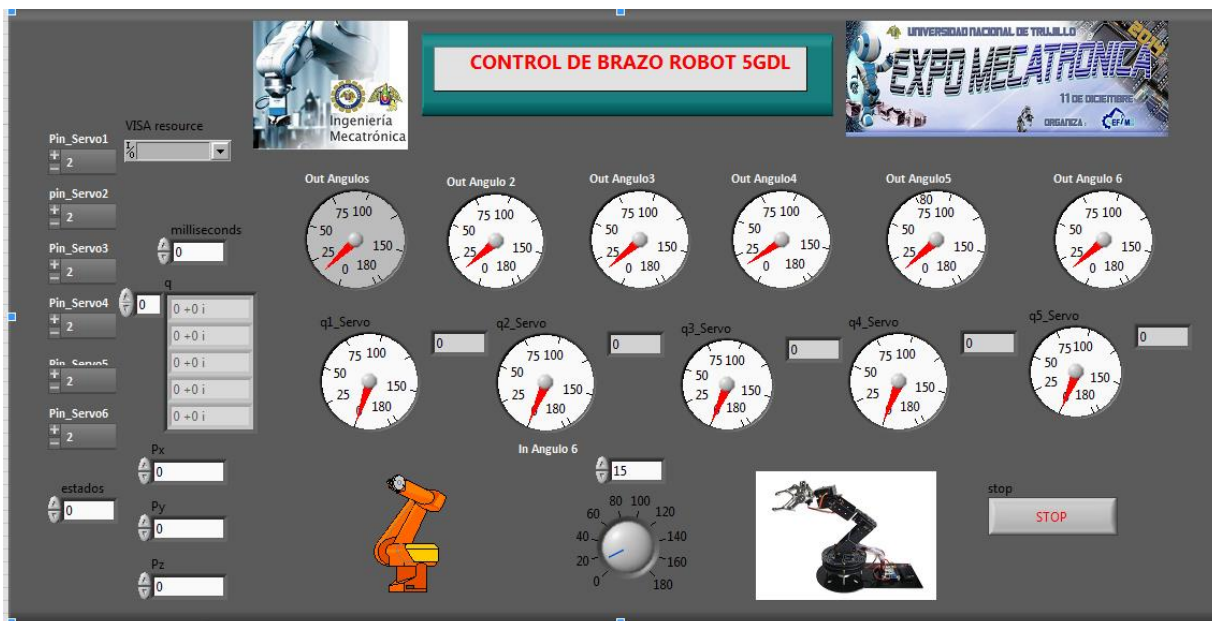


Figura 3.14 interfaz en labview.

3.8 FUNCIONAMIENTO.

EL funcionamiento del Brazo Robot 5GDL se mostrara en un video que adjunta con este trabajo.

4 .- BIBLIOGRAFIA

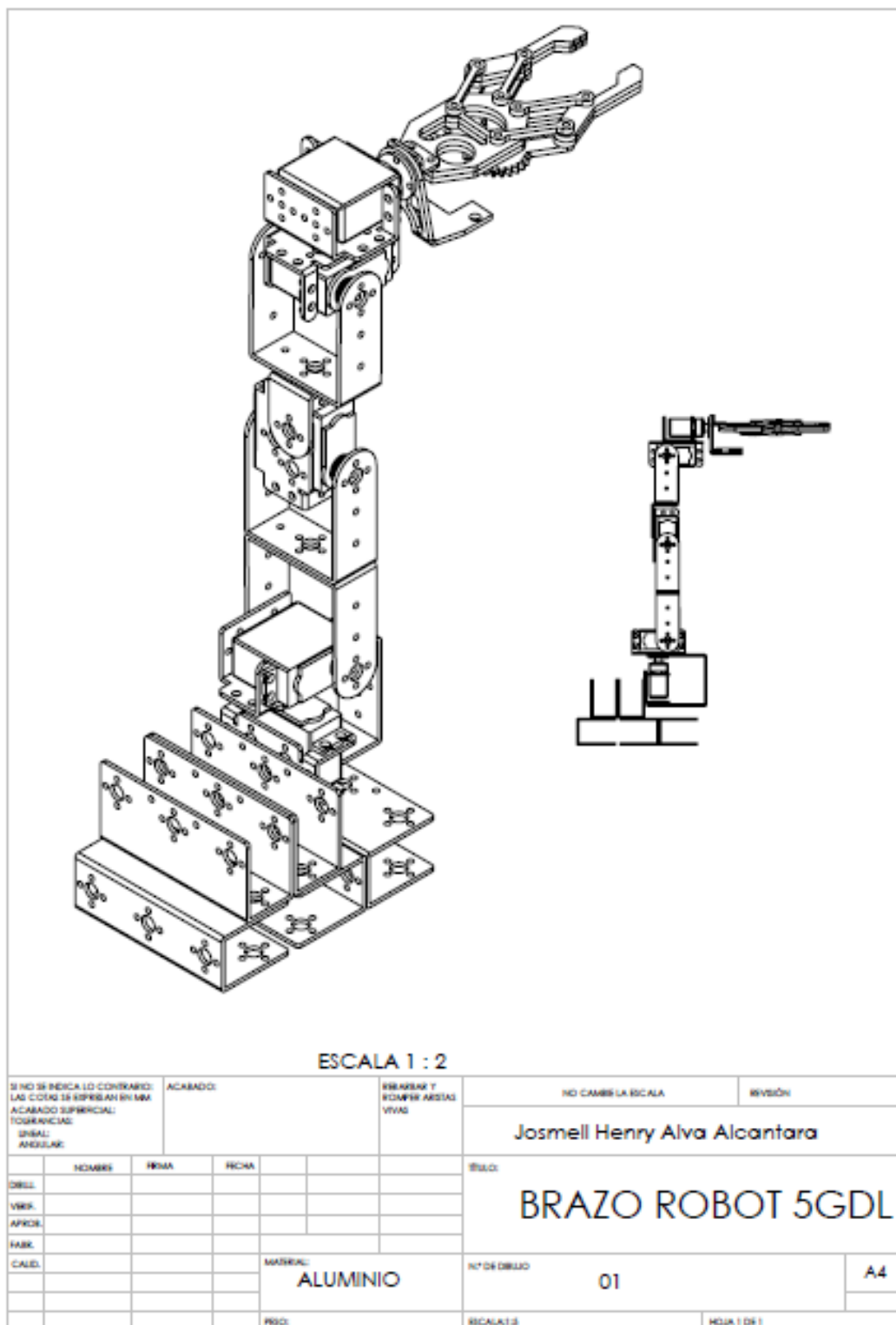
[1] BARRIENTOS Antonio, *Fundamentos de Robótica*, McGraw W – hill/interamericana de España 1997, ISBN: 84-481-0815-9

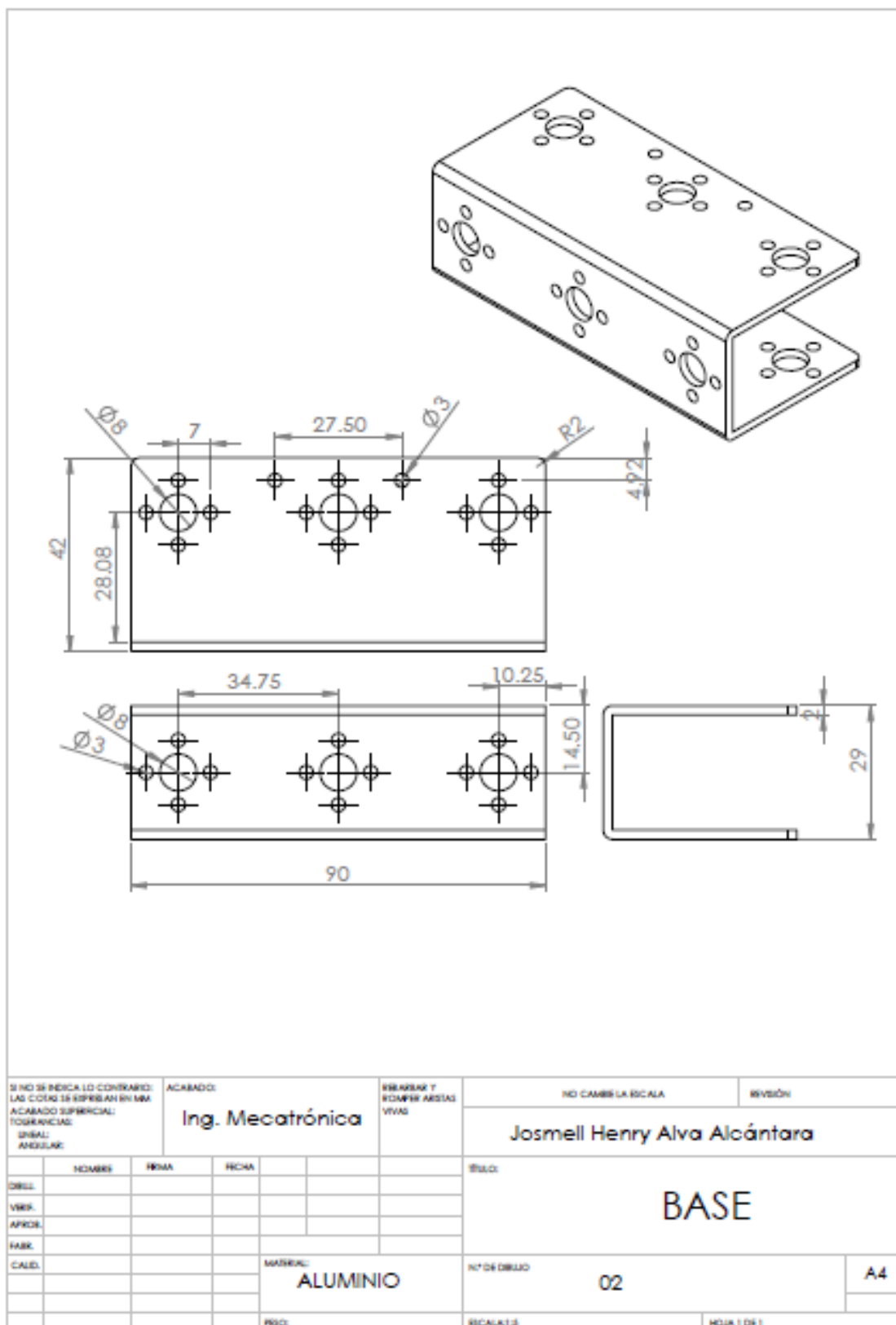
[2] FU, GONZALEZ, LEE; *ROBOTICA Control, detección, Vision e Inteligencia*. Sebastian Dormido Bencomo (tradu), McGraw W – hill/interamericana de España 1988, ISBN: 0-07-022625-3

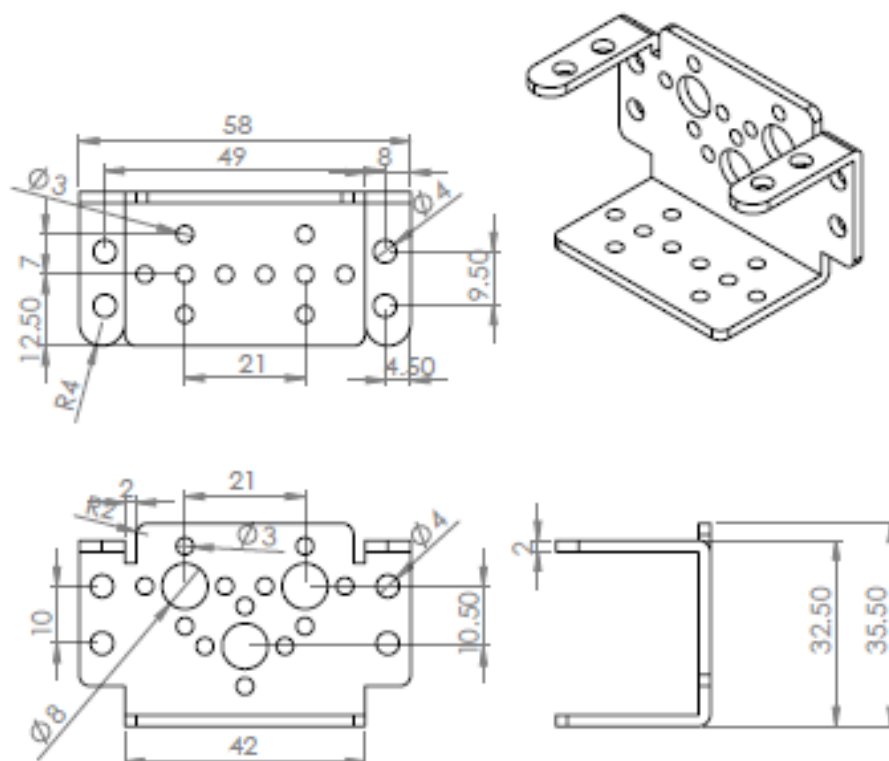
[3] KATSUHIKO OGATA, *Ingeniería de Control Moderno*, Person Educación S.A 2010 Madrid España, ISBN : 978-84-8322-660-5

5.- ANEXOS.

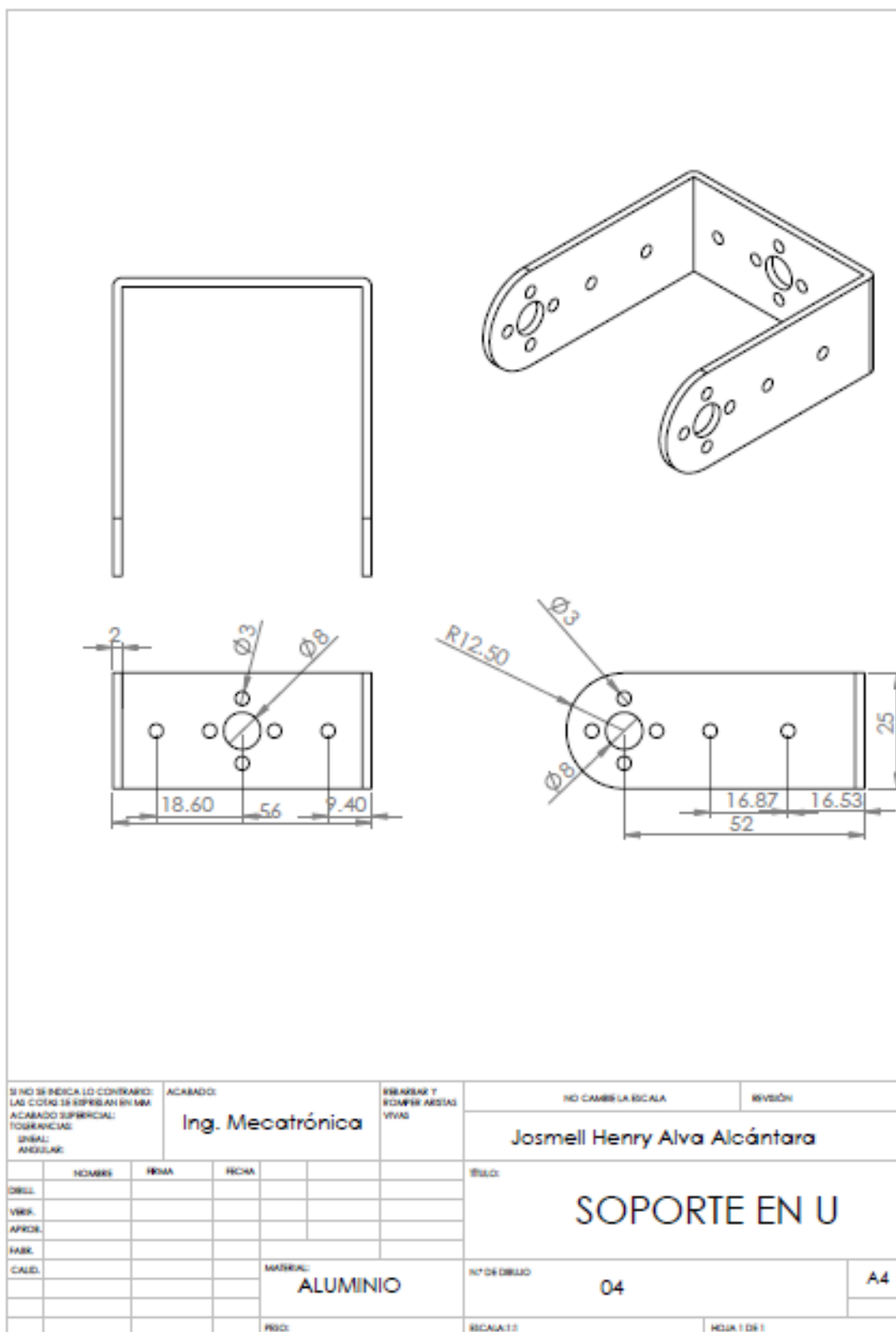
PLANOS DEL BRAZO ROBOT 5GDL – SOLIDWORKS.

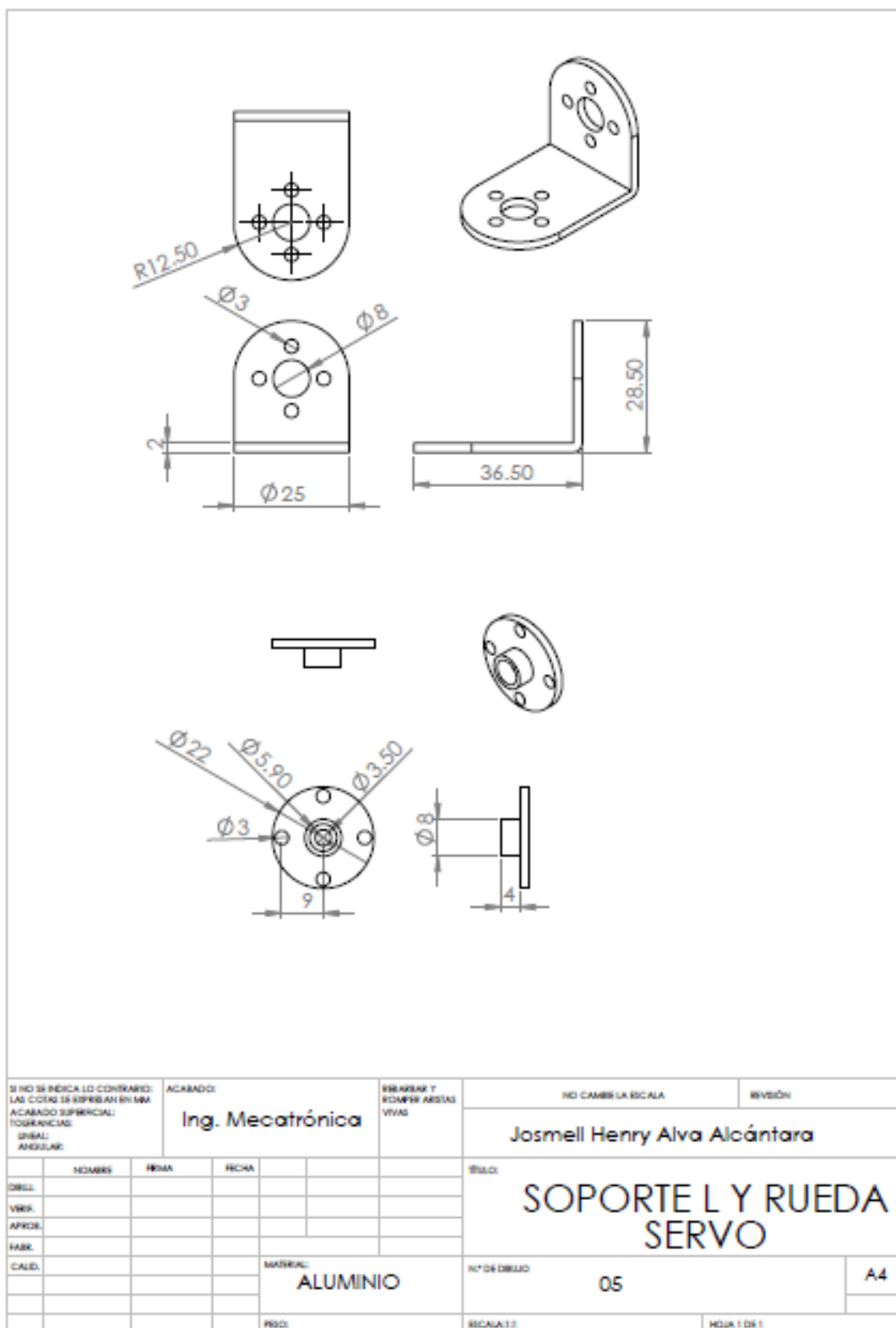






SI NO SE INDICA LO CONTRARIO: LAS COTAS SE EXPRESAN EN MM ACABADO SUPERFICIAL: TOLERANCIAS: LINEAL: ANGULAR:				ACABADO: Ing. Mecatrónica		REBARBAR Y ROMPER ARISTAS VIVAS		NO CAMBIA LA ESCALA		REVISIÓN	
								Josmell Henry Alva Alcántara			
								TÍTULO: SOPORTE			
								Nº DE DIBUJO 03			
								A4			
								HOJA 1 DE 1			





<p>SI NO SE INDICA LO CONTRARIO: LAS COTAS SE EXPRESAN EN MM ACABADO SUPERFICIAL: TOQUE FINO LAS: UNITS: ANGULARES:</p>		ACABADO:		REBARBAR Y ROMPER ARISTAS VIVAS		NO CAMBIE LA ESCALA		REVISIÓN	
		Ing. Mecatrónica						Josmell Henry Alva Alcántara	
								TÍTULO:	
								GARRA SUJECCION	
								Nº DE DISEÑO	
								06	
								A4	
								HOJA 1 DE 1	

BRAZO ROBOT 5GDL – EN LA IV EXPOMECATRÓNICA.

