

# DISEÑO DE UN ROBOT ANTROPOMÓRFICO DE PROPÓSITO GENERAL

## DOCUMENTO DE PROYECTO

TL. CARLOS ANDRES RODRIGUEZ PEREZ

DIRECTOR: M.Sc. William Prado Martinez

UNIVERSIDAD TECNOLÓGICA DE PEREIRA  
FACULTAD DE TECNOLOGÍAS  
INGENIERÍA MECATRÓNICA POR CICLOS

2013

Nota de Aceptación:

---

---

---

---

---

Firma del Jurado

---

Firma del Jurado

---

## **DEDICATORIA**

A Dios, que hace todo posible.

A mi esposa Carolina, por su apoyo y esfuerzo. Ha sido quien en gran medida me dado todo lo necesario y que sin tu ayuda no hubiese podido realizar este proyecto.

A Consuelo, quien siempre ha estado pendiente y al tanto de este proyecto, sin su colaboración, este proceso hubiese sido mas difícil.

A mis hijos, motor fundamental de mi vida, quienes día a día, me renuevan de energía para seguir adelante.

A mis padres, que son la base, y me han guiado por la vida.

## AGRADECIMIENTOS

Al ingeniero, William Prado por ser compañero, colega y amigo.

A los ingenieros Osiel Arbelaez y Jairo Mendoza, que me han brindado todo su apoyo.

Al ingeniero Wilson Perez, que ha sido compañía y apoyo en la elaboración de este proyecto, que hizo posible encontrar la salida cuando parecía que no existía, ha demostrado que realmente existen los amigos.

## **RESUMEN**

En este documento de consigna el proceso de ingeniería inversa realizado al brazo robótico Kawasaki RS03N existente en los laboratorios de mecatrónica, de la universidad tecnológica de Pereira. Se describe el proceso de desarrollo de cada una de las partes del robot, con la ayuda de un software CAD y como este permite la obtención de los parámetros necesarios para el análisis tanto cinemático como dinámico el sistema. Se determina la cinemática directa del robot a través del método Denavit-Hartenberg, para con la ayuda del robotics toolbox de matlab realizar el estudio dinámico del sistema. Finalmente se desarrolla una interfaz de interacción con el robot ,con la ayuda de una GUI, que permitirá una interacción sencilla con el brazo robótico.

# Índice

<b>1. MORFOLOGÍA DEL ROBOT</b>	<b>12</b>
1.1. MÉTODO DENAVIT-HARTENBERG . . . . .	14
<b>2. DISEÑO DEL BRAZO ROBÓTICO</b>	<b>18</b>
2.1. OBTENCION DE PARAMETROS DEL ROBOT . . . . .	22
2.2. CINEMÁTICA DEL ROBOT . . . . .	23
<b>3. DINÁMICA DEL BRAZO ROBÓTICO</b>	<b>30</b>
3.1. ANÁLISIS DINÁMICO INVERSO . . . . .	31
<b>4. INTERFAZ DE CONTROL DEL ROBOT</b>	<b>33</b>
<b>5. CONCLUSIONES Y RECOMENDACIONES</b>	<b>37</b>
<b>Referencias</b>	<b>39</b>
<b>A. ANEXO 1: DIAGRAMA DE BLOQUES EN SIMULINK</b>	<b>40</b>
<b>B. ANEXO 2: CÓDIGO DE DESARROLLO DE LA INTERFAZ GRÁFICA</b>	<b>42</b>
<b>C. ANEXO3: ARCHIVOS DE CONSTRUCCIÓN EN INVENTOR 2013</b>	<b>54</b>

# Índice de figuras

1.1.	Comparación robot antropomórfico[2] . . . . .	12
1.2.	Cadena cinemática[5] . . . . .	13
1.3.	Parámetros del primer eslabón . . . . .	15
1.4.	Parámetros segundo eslabón . . . . .	16
1.5.	Parámetros tercer eslabón . . . . .	17
2.1.	Robot Kawasaki RS03N . . . . .	18
2.2.	Base. . . . .	19
2.3.	Hombro . . . . .	19
2.4.	Codo . . . . .	20
2.5.	Muñeca . . . . .	20
2.6.	Rotador . . . . .	21
2.7.	Girador . . . . .	21
2.8.	Vista isometrica brazo robotico Kawasaki RS03N . . . . .	22
2.9.	Ejes de rotación del robot . . . . .	23
2.10.	Complemento SimMechanics en Inventor . . . . .	24
2.11.	Diagrama de bloques del ensamble . . . . .	25
2.12.	Creación de los eslabones . . . . .	25
2.13.	Definición de las masas de los eslabones . . . . .	25
2.15.	Centros de masa . . . . .	26
2.14.	Inercias de los eslabones . . . . .	26
2.16.	Enlace de los eslabones . . . . .	26
2.17.	Eslabones en estructura de alambre . . . . .	27
2.18.	Posiciones definidas . . . . .	27
2.19.	Análisis cinemática directa . . . . .	28
2.20.	Posición del efecto final . . . . .	28
2.21.	Análisis cinemática inversa . . . . .	28
2.22.	Movimiento necesario . . . . .	29
2.23.	Generación de trayectorias . . . . .	29
2.24.	Posición p1 de brazo . . . . .	29
3.1.	Generación de trayectorias . . . . .	30
3.2.	análisis dinámico inverso . . . . .	31
3.3.	Torques requeridos . . . . .	31
4.1.	Conexión subsistemas . . . . .	33
4.2.	Detalle bloque subsistema . . . . .	33
4.3.	Modelo 3D en matlab . . . . .	34
4.4.	Pantalla de Inicio . . . . .	35
4.5.	Movimiento de las articulaciones . . . . .	35
4.6.	Simulación predefinida . . . . .	36
A.1.	Diagrama de bloques inicial Simulink . . . . .	40
A.2.	Diagrama de bloques final Simulink . . . . .	41
B.1.	Código matlab . . . . .	42

B.2.	Simulación libre . . . . .	43
B.3.	Simulación libre . . . . .	44
B.4.	Simulación libre . . . . .	45
B.5.	Simulación libre . . . . .	46
B.6.	Simulación predefinida . . . . .	47
B.7.	Simulación predefinida . . . . .	48
B.8.	Simulación predefinida . . . . .	49
B.9.	Simulación predefinida . . . . .	50
B.10.	Simulación predefinida . . . . .	51
B.11.	Simulación predefinida . . . . .	52
B.12.	Simulación predefinida . . . . .	53

## **Índice de cuadros**

1. Parámetros dinámicos del robot . . . . .	22
2. Parametros Denavit-Hartenberg . . . . .	24

# INTRODUCCIÓN

Cada vez es mayor el número de robots industriales que se usan en las empresas alrededor del mundo, en especial en líneas de producción automatizadas, donde los robots juegan un papel importante en la ejecución de tareas complejas y de alta repetibilidad.

Los principales países usuarios de robots son Estados Unidos, Japón, Alemania, Italia, Francia y China. Para el año 2008 se encontraban operando en todo el mundo más de un millón de robots industriales.[4]

Colombia se encuentra en una etapa de oportunidades, transición y adaptación; el país cuenta con una industria todavía en desarrollo, que se apresta a hacerle frente a un nuevo paradigma “de pasar de exportador de materias primas sin transformación a dar valor agregado a sus productos”.

Para lograrlo, es necesario modernizar la estructura productiva, la maquinaria y el equipamiento de capital, para poder competir en los mercados mundiales, no sobre la base de salarios bajos, sino de mayor valor agregado y de complejidad tecnológica. Desde este punto de vista, Los robots industriales ocupan un lugar destacado dentro del mejoramiento y competitividad del aparato productivo.

A través de este proyecto se pretende comprender todo lo relacionado con el diseño de robots, con la intención de poder ofrecer soluciones adecuadas a cada una de las necesidades del sector productivo del país, y así poder fomentar una mejora tecnológica significativa para la industria Colombiana.

El programa de ingeniería mecatrónica de la universidad tecnológica de Pereira, posee en el laboratorio de automatización industria un brazo robótico Kawasaki RS03N, el cual ofrece la capacidad realizar actividades de soldadura de arco, sellado, dispensación, manipulación y remoción de material, ensamble e inspección. Posee una unidad de control serie E, para robots de pequeña escala y un panel de programación portátil, que permite la programación del robot por parte del usuario.

Este tipo de robots tienen un amplio uso en el sector industrial, ya que su robustez y amplias capacidades de operación, permiten la ejecución de muchas actividades que son de riesgo para las personas o que por la complejidad de los procesos se tornan difíciles para un operario. Esto favorece tiempos de producción más cortos y en consecuencia un incremento de la productividad a nivel de las empresas.

El movimiento del robot está basado en complejos algoritmos de cinemática, super-

visada por una unidad de control central electrónica, dotado de una robustez que le permite ser usado en ambientes industriales severos.

Estos robots se operan a través de un lenguaje de programación estructurado que permite el control de todo el ciclo automático, archivar datos de operación, modificar y supervisar los parámetros de la máquina.

El objetivo de este proyecto es proponer un diseño de un robot antropomórfico de propósito general y sus alcances son proponer el modelo dinámico y el diseño de sólido 3D , el diseño de una interfaz de programación del brazo robótico y proponer un sistema de control.

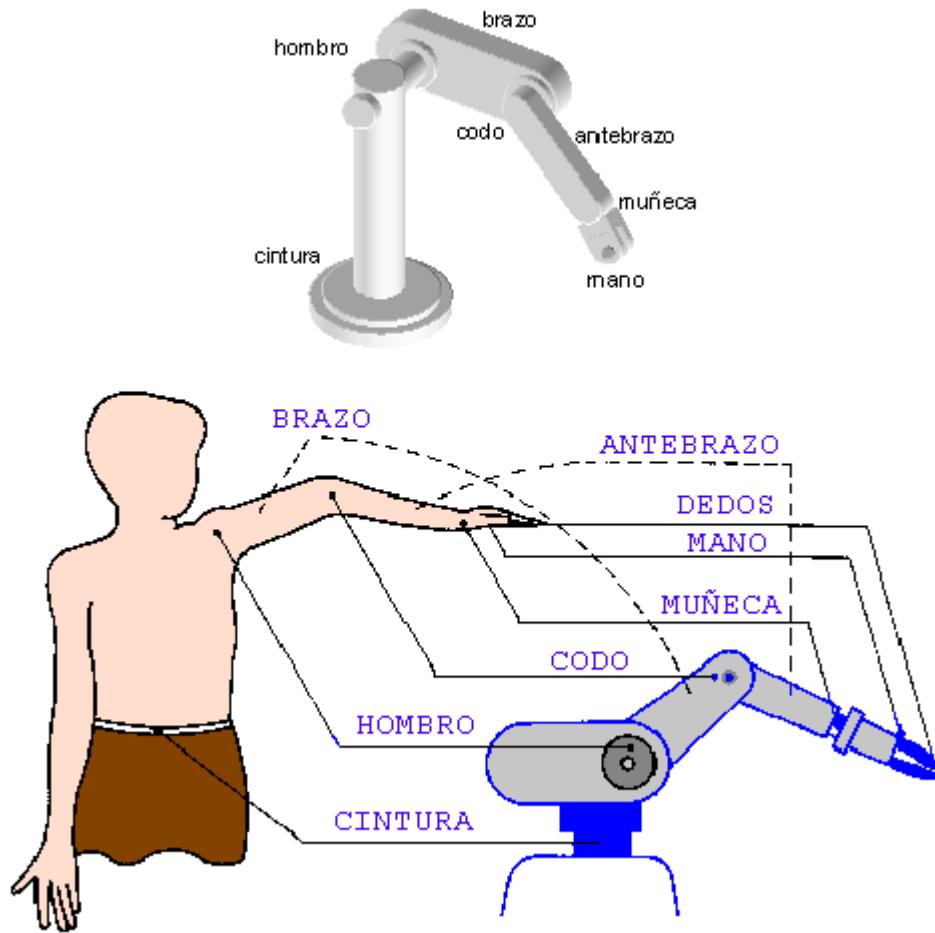
Para el desarrollo de este proyecto se trabajará sobre la base del robot Kawasaki RS03N. A través de técnicas de ingeniería inversa, se determinarán las características constructivas , inventario de componentes; apoyados en la construcción del diseño en 3D en un software CAD; para poder establecer el modelo matemático del sistema dinámico y poder reproducir el comportamiento del robot, lo que permitirá determinar un modelo general para el desarrollo de robots antropomórficos.

Se desarrollara una interfaz de trabajo través de matlab, permitiendo la programación y ajustes de operación del brazo robótico y se explorarán estrategias de control adecuada para lograr el correcto funcionamiento y operación del brazo.

# 1. MORFOLOGÍA DEL ROBOT

Dependiendo de tipo de articulaciones que posee un robot, se puede definir su clasificación, existen articulaciones rotacionales (que generan solamente movimiento de rotación) y prismáticas o lineales ( que generan desplazamientos longitudinales) y el conjunto de estas puede definir el tipo de robot industrial entre Antropomórfico, (ver figura1.1), (con mínimo 3 articulaciones rotacionales), SCARA (dos articulaciones rotacionales y una prismática), Esférico (dos articulaciones rotacionales y una prismática), Cilíndrico (una articulación rotacional y dos prismáticas ) y Cartesiano (tres articulaciones prismáticas).

Figura 1.1: Comparación robot antropomórfico[2]

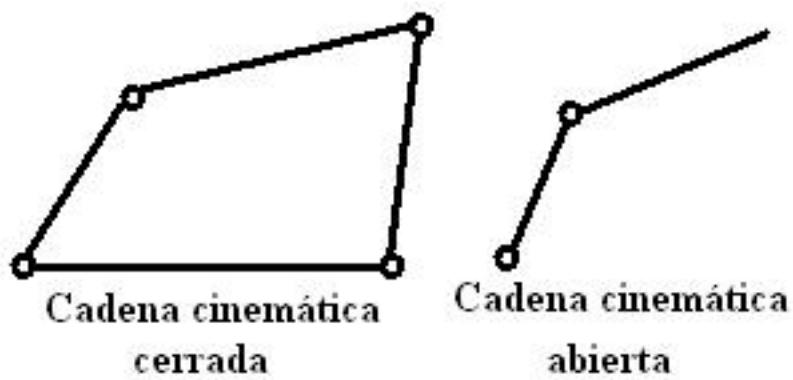


El posicionamiento del robot en el espacio tridimensional requiere de 6 coordenadas (tres para la posición cartesiana y 3 para la orientación de la herramienta de trabajo), la relación establecida entre coordenadas cartesianas, articulares y su orientación se

denomina cinemática directa.

Cuando se habla de morfología del robot se hace referencia a la descripción de los componentes, partes y estructura mecánica, básicamente un robot es un sistema que permite realizar una gran variedad de actividades, como traslado de objetos, operaciones de pintura, procesos de soldadura, ensamble de piezas, entre otras. En esencia un robot esta compuesto por una serie consecutiva de eslabones y articulaciones que forman una cadena cinemática abierta (ver figura 1.2). Una cadena cinemática abierta se caracteriza por que los dos extremos de la cadena no se unen, es decir no forman un lazo cerrado, de lo contrario sería una cadena cinemática cerrada.

Figura 1.2: Cadena cinemática[5]



Un robot esta constituido por una primera articulación que conforma la base; seguida por una sucesión de eslabones y articulaciones y al final hay espacio para colocar una herramienta, que permite llevar a cabo determinadas labores, lo que indica que el extremo final no esta conectado a la base. Cada una de las articulaciones define un grado de libertad y son las encargadas de transmitir la energía para producir movimiento a cada uno de los eslabones que se haya conectado, los eslabones se constituyen de barras en teoría rígidas acopladas mecánicamente al rotor de una articulación y al estator de la siguiente.

El espacio de trabajo de un robot, esta definido como el lugar donde el sistema puede realizar todos sus posibles movimientos, determinado por la geometría del robot y el tipo de articulaciones que lo conforman.

La representación de posicionamiento del robot implica sistema coordenado cartesiano que definen la posición y orientación del extremo final de robot; la transformación homogénea es una herramienta que relaciona las operaciones de rotación y traslación dentro de una matriz que permite el análisis de la cinemática directa. La transformación

homogénea esta dada por la expresión 1.1.

$$H = \begin{bmatrix} R_{z,\theta} & d_0^1 \\ 0^T & 1 \end{bmatrix} \quad (1.1)$$

Donde  $R$  representa la matriz de rotación ortogonal,  $d$  representa el vector de traslación,  $0$  es una matriz de acoplamiento para ajuste de dimensiones y en la ultima fila y columna aparece el numero 1.

La Transformación inversa esta determinada por:

$$H^{-1} = \begin{bmatrix} R_{z,\theta}^T & -R_{z,\theta}^T d_0^1 \\ 0^T & 1 \end{bmatrix} \quad (1.2)$$

La matriz de rotación  $R$  aparece como transpuesta, esto es debido a que la inversa de una matriz ortogonal coincide con su transpuesta.

las matrices de transformación permiten modelar la posición de la herramienta, que presentadas en una sola matriz, contienen la posición y orientación de la herramienta, permitiendo el modelo cinemático directo[4]

La cinemática directa permite un mejor entendimiento del posicionamiento del robot, ya que permiten la transformación de las coordenadas articulares en coordenadas cartesianas, mucho mas comprensibles para la persona que vaya a trabajar con robots.

## 1.1. MÉTODO DENAVIT-HARTENBERG

Existen múltiples métodos para obtener la cinemática directa de un modelo mecánico, por ejemplo por geometría aplicada, esta herramienta es útil cuando el numero de grados de libertad es limitado, ya que para grados de libertad mayores el planteamiento se torna complejo; el método Denavit\_Hartenberg ofrece una herramienta sencilla para obtener la cinemática directa en representación de transformaciones homogéneas.

El procedimiento para la implementación del método corresponde de manera inicial a determinar los ejes de rotación numerados de  $z_0, z_1, \dots, z_{n-1}$ .

Luego se establece el sistema de referencia cartesiano fijo  $R_0(x_0, y_0, z_0)$  en la base del robot, los ejes  $x_0, y_0$  son definidos por regla de la mano derecha. Los sistemas de referencia de los elementos subsiguientes se definen desde  $R_{i-1}(x_{i-1}, y_{i-1}, z_{i-1})$  hasta  $R_i(x_i, y_i, z_i)$ .

Seguidamente se determina el origen de cada sistema de referencia  $O_i$  en la intersección normal que une a  $z_{i-1}$  con  $z_i$ .  $x_i$  se define a lo largo de la normal que une a  $z_{i-1}$  y  $z_i$ , en dirección de  $i-1$  a  $i$ .

Finalmente se establece el sistema de referencia del extremo final  $R_n(x_n, y_n, z_n)$ .

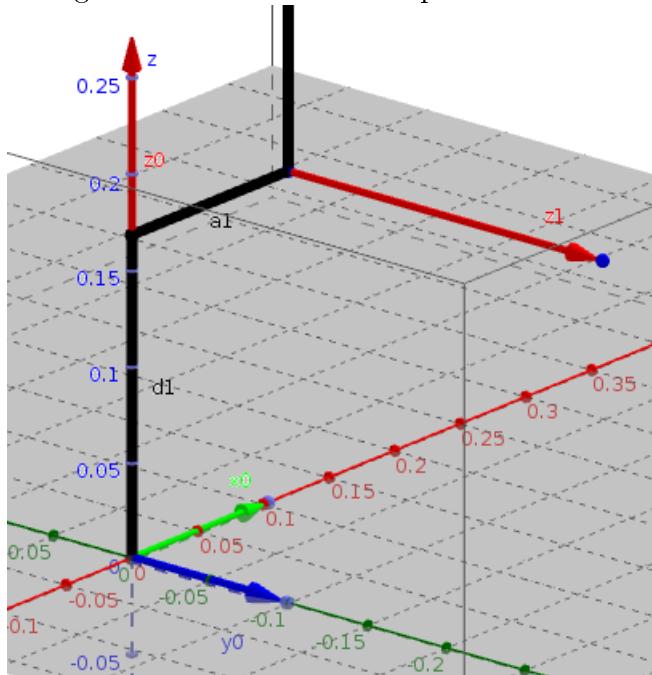
Para establecer los parámetros se define la siguiente convención:

1.  $a_i$  es la longitud entre  $z_{i-1}$  y  $z_i$  a lo largo de  $x_{i-1}$ .
2.  $\alpha_i$  es el ángulo medido entre  $z_{i-1}$  a  $z_i$ .
3.  $d_i$  es la distancia entre  $x_{i-1}$  a  $x_i$  a lo largo del eje  $z_{i-1}$ .
4.  $\theta_i$  corresponde al desplazamiento rotacional entre  $x_{i-1}$  a  $x_i$  alrededor de  $z_{i-1}$ .

Sobre estas bases, se procede a la determinación de los parámetro Denavit-Hartenberg, tomando las medidas reales del robot, cabe aclarar que en los manuales del brazo no aparecen las dimensiones del mismo, por lo tanto se procede a tomar medidas con la ayuda de cinta métrica y un pie de rey, con la intención de obtener los valores lo mas cercanos a la realidad, para proseguir de la siguiente manera:

Se define como origen la base del robot, en el punto donde el primer eslabón hace contacto con la base, como se muestra en la siguiente figura (ver figura 1.3).

Figura 1.3: Parámetros del primer eslabón

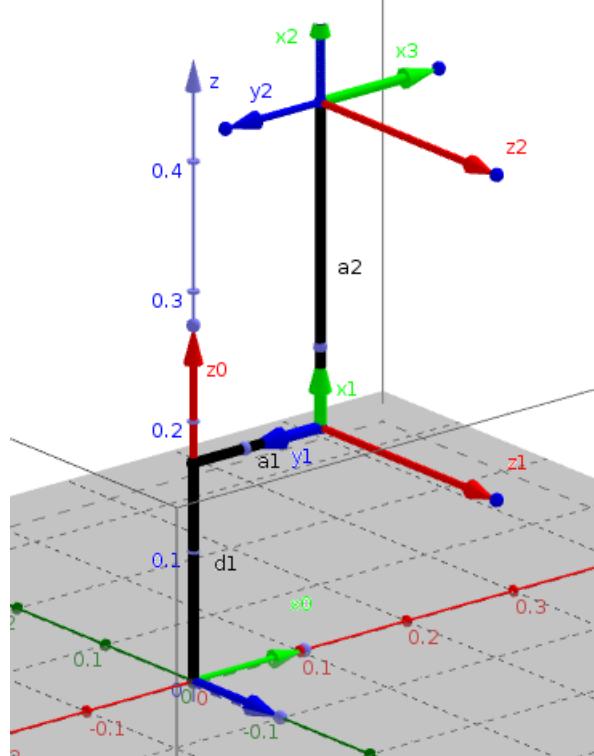


$z_0, x_0, y_0$  se ubican en la base del robot,  $z_1$  corresponde a la articulación del hombro, con esto se tiene entonces que  $a_1=0.119m$ ,  $d_1=0.168m$ ,  $\alpha_1=\pi/2$  y  $\theta_1=\theta_1$ .

El siguiente eslabón contiene las articulaciones hombro y codo (ver figura 1.4), los parámetros obtenidos son los siguientes:

$a_2=0.250m$ ,  $d_2=0$ ,  $\alpha_2=0$  y  $\theta_2=\theta_2 - \frac{\pi}{2}$ . El  $\pi/2$  se resta ya que  $x_1$  se encuentra rotado con respecto a  $x_0$ .

Figura 1.4: Parámetros segundo eslabón



El tercer eslabón contiene las articulaciones codo y muñeca, de acuerdo a la convención D-H cuando dos ejes de rotación se interceptan, el origen se ubica sobre la intersección de dichos ejes, (ver figura 1.5).

Por lo tanto  $a_3=0$ ,  $d_3=0$ ,  $\alpha_3=-\pi/2$  y  $\theta_3=\theta_3$ .

El cuarto eslabón contiene las articulaciones muñeca y rotador, al igual que en el caso anterior los ejes de estas articulaciones se interceptan, por lo tanto el origen de  $z_4$  y  $z_5$  es coincidente.

Por lo tanto  $a_4=0$ ,  $d_4=0.223m$ ,  $\alpha_4=\pi/2$  y  $\theta_4=\theta_4$ .

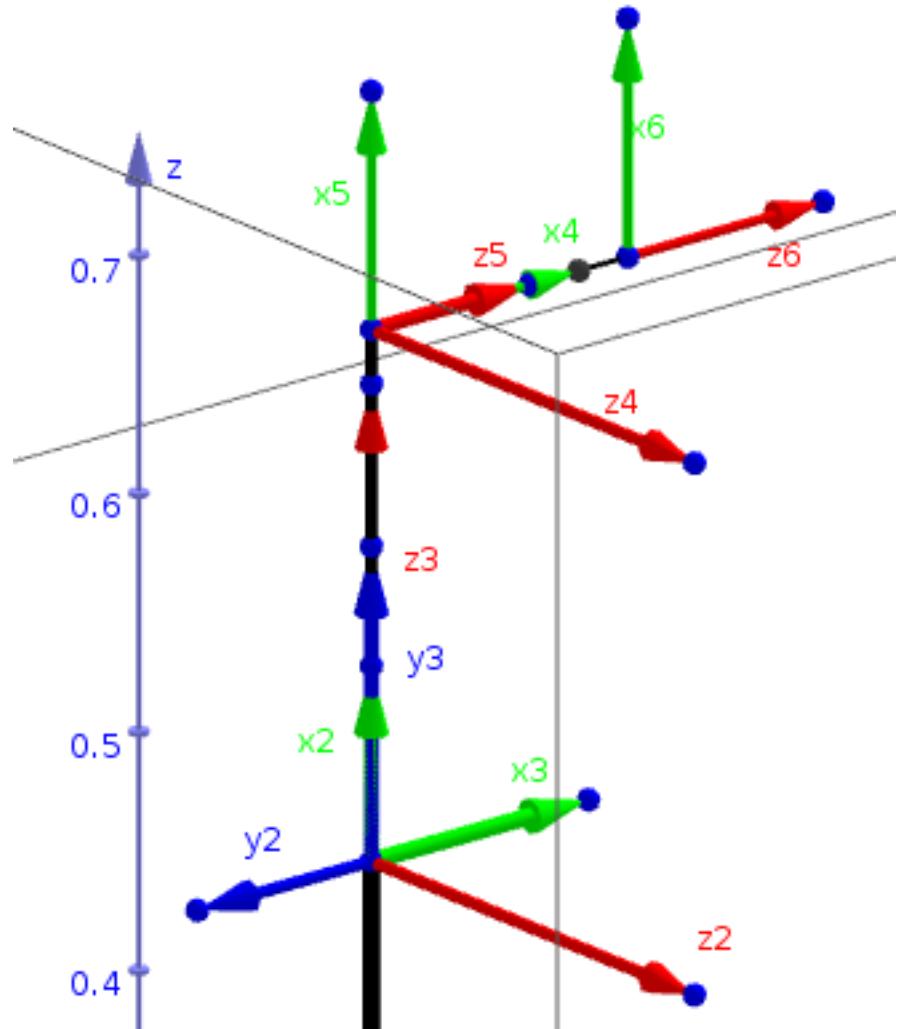
Para el quinto eslabón los parámetros son:  $a_5=0$ ,  $d_5=0$ ,  $\alpha_5=\pi/2$  y  $\theta_5=\theta_5 + \frac{\pi}{2}$ .

Para el sexto eslabón los parámetros son:  $a_6=0$ ,  $d_6=0$ ,  $\alpha_6=0.07m$  y  $\theta_6=\theta_6$ .

Estos parámetros son los usados para la creación del cuadro 2 y la definición de los

valores en el robotics toolbox.

Figura 1.5: Parámetros tercer eslabón



## 2. DISEÑO DEL BRAZO ROBÓTICO

Tomando como punto de partida el brazo robótico Kawasaki RS03N (ver figura 2.1), del programa de ingeniería mecatrónica, se procede a realizar las mediciones con la intención de obtener las características constructivas del mismo, para poder proceder a la construcción de un modelo con la ayuda de un software CAD.

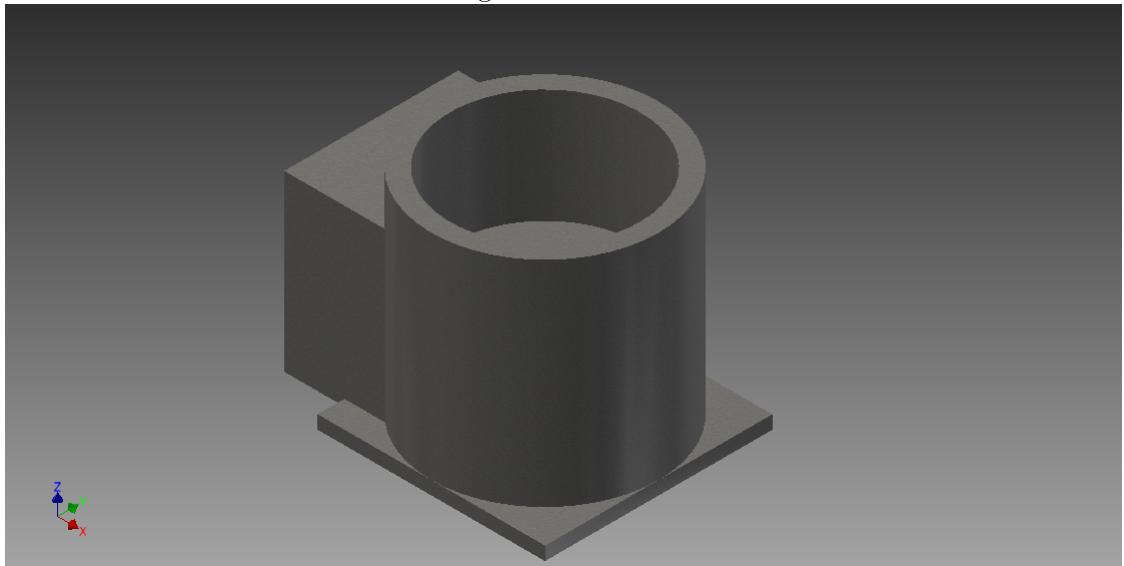
Figura 2.1: Robot Kawasaki RS03N



Este brazo robótico cuenta con 6 grados de libertad definidos por articulaciones de tipo rotacional, que son motorizadas a través de 6 servomotores. Este brazo esta constituido por seis eslabones que serán definidos de la siguiente manera:

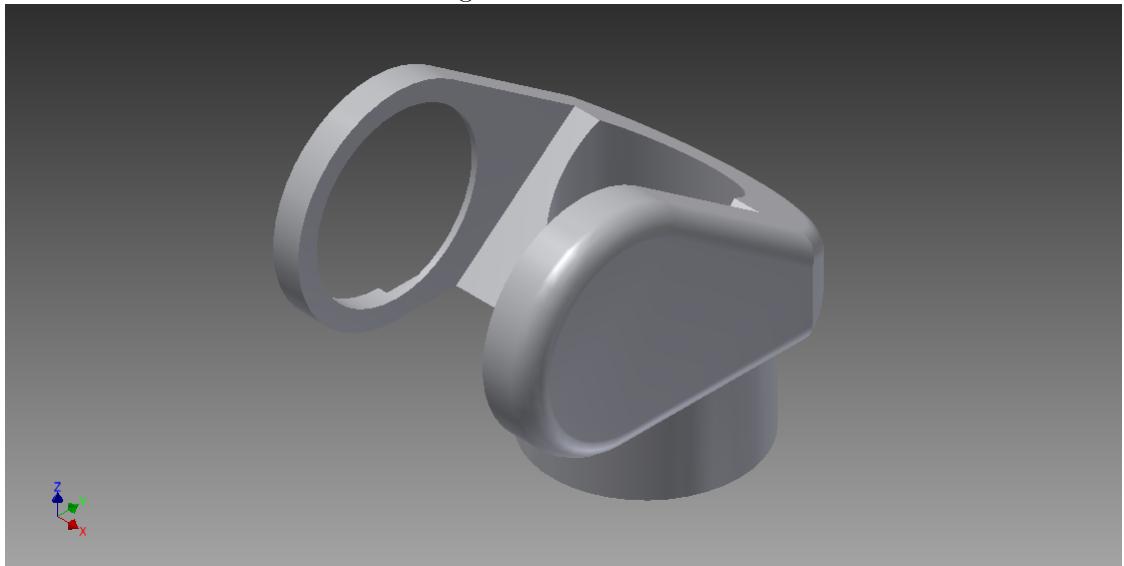
- Base: Constituye el soporte de todo el sistema (ver figura 2.2), definido como eslabón fijo, al rededor del cual se mueven los demás elementos del brazo, este elemento posee la primer articulación.

Figura 2.2: Base.



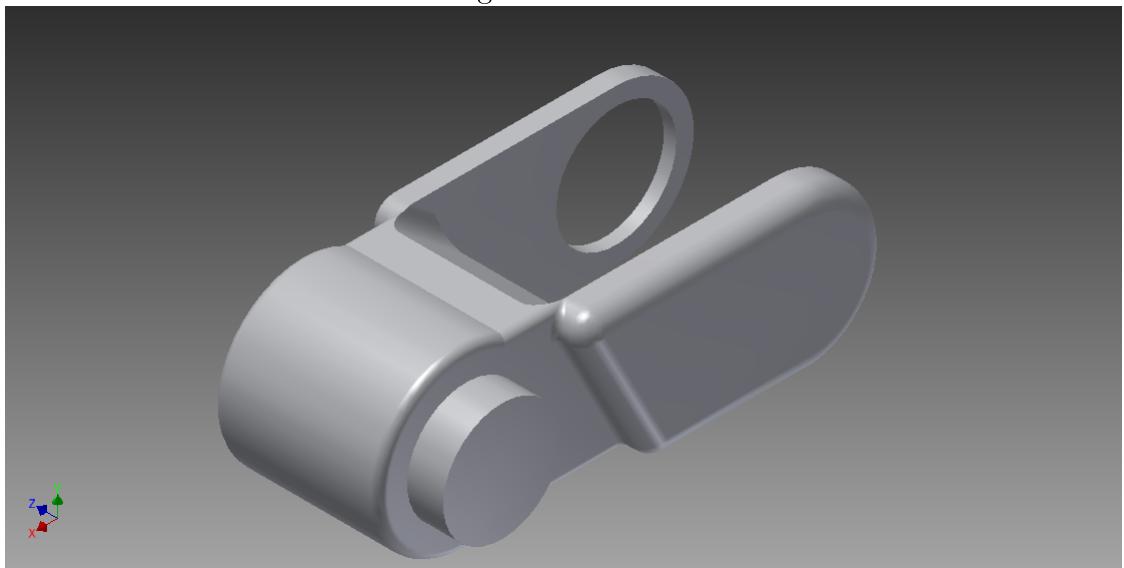
- Hombro: Constituye el segundo eslabón del conjunto (ver figura 2.3), con una articulación de tipo rotacional, que por su ubicación es el elemento que mas trabajo debe desarrollar de todos los eslabones del sistema, ya que sobre su servo se ejercen fuerzas producto del resto de los eslabones del brazo.

Figura 2.3: Hombro



- Codo: Constituye el cuarto eslabón del sistema, (ver figura 2.4), posee una articulación de tipo rotacional.

Figura 2.4: Codo



- La muñeca esta constituida por tres ejes rotacionales, uno en la muñeca (ver figura2.5) y otro en el rotador (ver figura2.6) y otro en el girador (ver figura2.7).

Figura 2.5: Muñeca

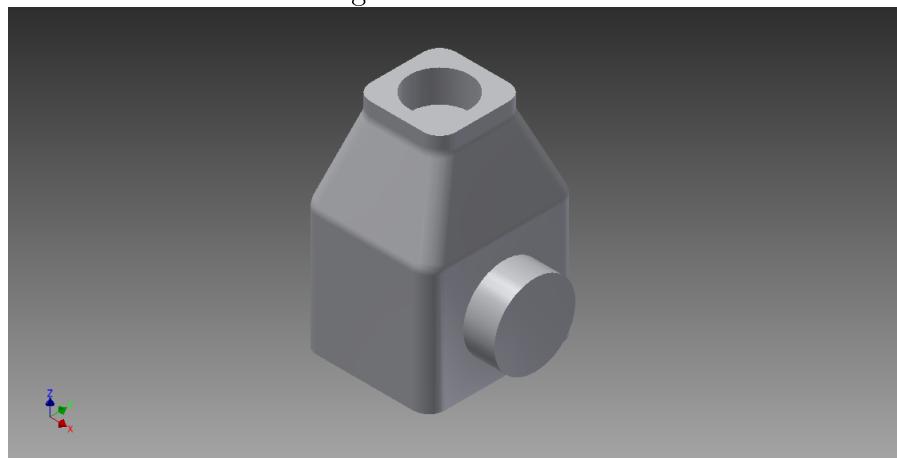


Figura 2.6: Rotador

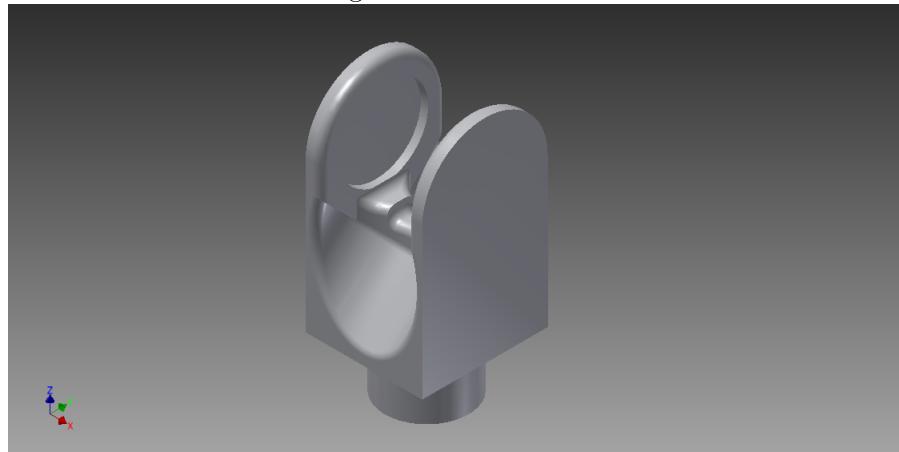
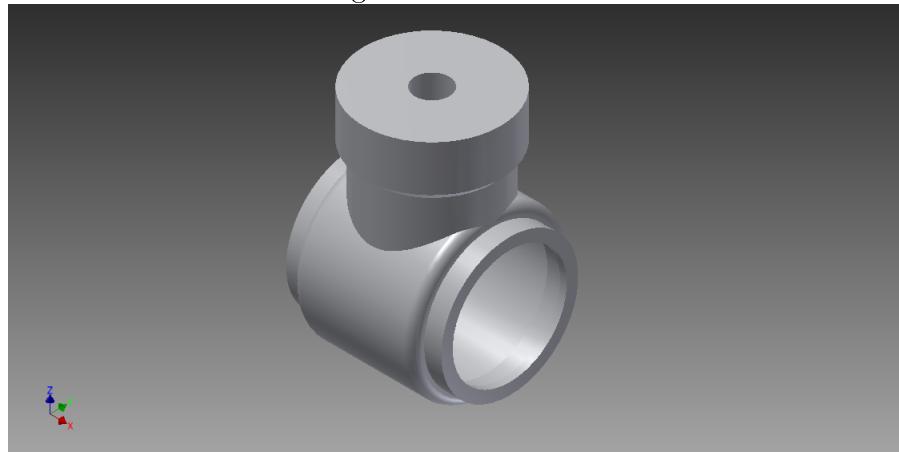


Figura 2.7: Girador

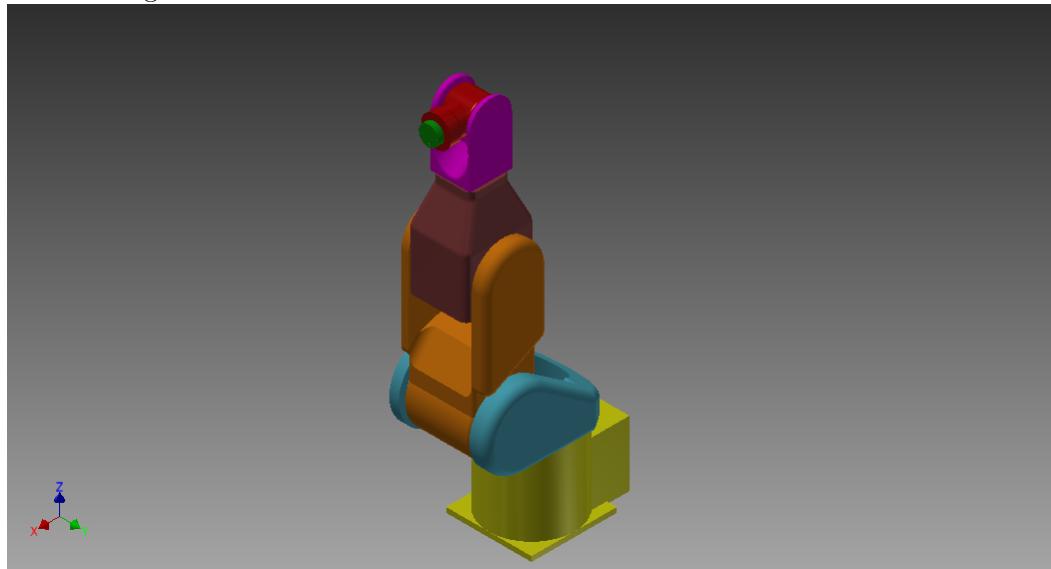


Para la construcción del modelo 3D se usó Autodesk inventor 2013 con una licencia educativa otorgada por la empresa Autodesk de manera gratuita, por un periodo de 3 años. Este software ofrecen un entorno de trabajo potente, que permite desde realizar el análisis de resistencia de materiales, hasta el análisis dinámico del sistema. En el anexo 3 se encuentran los archivos de construcción del brazo.

Inventor, permite dentro de sus múltiples capacidades, determinar el material con el cual se construyen los elementos, para el desarrollo de este proyecto se trabajo con aluminio 6061, aluminio endurecido común de uso general, que ofrece una buena resistencia a las estructuras trabajadas con este material.

En la siguiente figura 2.8 se puede ver la proyección general del brazo con la identificación de sus grados de libertad en la posición de home definida por el autor.

Figura 2.8: Vista isométrica brazo robotico Kawasaki RS03N



## 2.1. OBTENCION DE PARAMETROS DEL ROBOT

Con el diseño del robot ya es posible la obtención de los parámetros físicos del robot, como su masa, momentos de inercia, y parámetros geométricos que permitirán establecer tanto el modelo cinemático y dinámico del robot.

A través de Inventor es posible determinar de manera sencilla cada uno de los parámetros antes definidos para cada uno de los eslabones, a partir de sus características constructivas y el material de cada uno de los eslabones. En el cuadro 1 se resumen los parámetros dinámicos del robot.

Cuadro 1: Parámetros dinámicos del robot

Eslabón	Masa (g)	Centros de gravedad (mm)			Momentos de inercia (Kg*mm <sup>2</sup> )					
		rx	ry	rz	I <sub>xx</sub>	I <sub>xy</sub>	I <sub>xz</sub>	I <sub>yy</sub>	I <sub>yz</sub>	I <sub>zz</sub>
Hombro	2712	23,819	14,537	164,384	18958,499	-2872,269	-5524,877	21516,366	-3147,684	20678,925
Codo	2465	119,357	128,164	345,758	34256,520	-174,057	197,368	29430,084	873,750	12195,699
Muñeca	2005	119,000	15,312	546,777	9621,012	0	0	9095,815	1570,643	6788,083
Rotador	1239	119,000	-0,750	677,920	2811,371	0	0	2528,956	0	1352,585
Girador	201	140,150	-0,750	747,039	165,212	0	0	320,542	0	306,573
P. herramienta	77	178,466	-0,750	747,039	9,150	0	0	54,431	0	54,431

Los datos de la tabla anterior permiten el proceso de análisis dinámico del sistema.

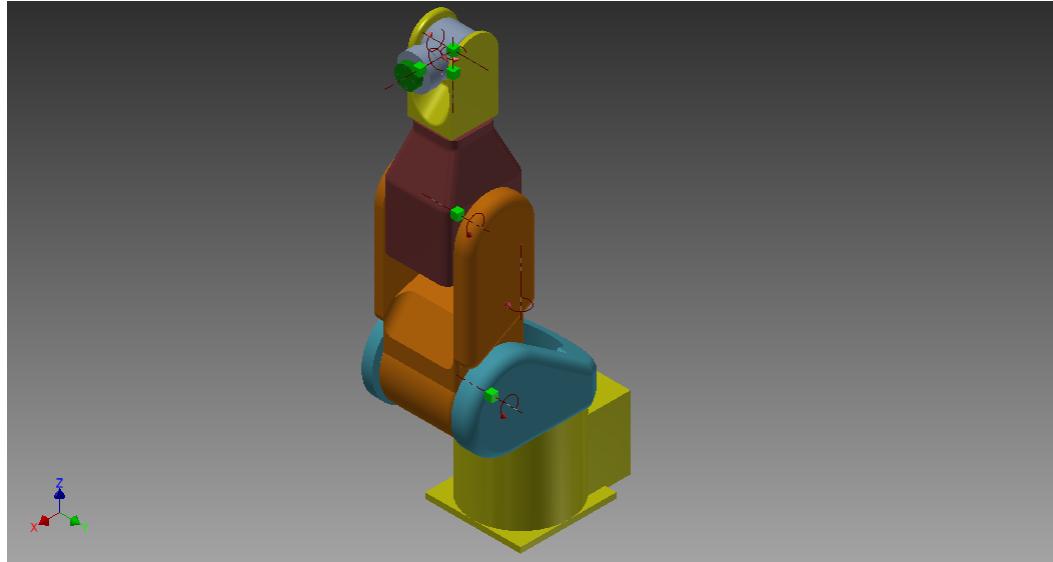
## 2.2. CINEMÁTICA DEL ROBOT

La cinemática permite relacionar las trayectorias articulares, con las coordenadas cartesianas del efecto final.

Para comenzar se deben definir los ejes de rotación de cada articulación y trazar una recta perpendicular o colineal según sea el caso que los una, ya que en las intersecciones se definirán los puntos de origen de cada articulación.

En la figura 2.9 se pueden observar los ejes de rotación y la rectas que los unen, esto se logra apoyado en el modelo 3D previamente desarrollado.

Figura 2.9: Ejes de rotación del robot



Para continuar con el proceso, se utilizará el método Denavit-Hartenberg que permite la obtener de manera sencilla los parámetros necesarios para la construcción de la matriz de transformación que permitirá determinar la cinemática directa del robot. Existen otros métodos para obtener la cinemática del robot pero dado el numero de grados de libertad que posee el mecanismo el método Denavit-Hartemberg los hace mas simple. El origen del sistema se define en la base del robot, cada eje de rotación se define como eje z y el eje x se toma como la linea que une  $z_{n-1}$  con  $z_n$  y el eje y se asume por la regla de la mano derecha.[4]

En la tabla 2 se observan los parametros Denavitt-Hartemberg obtenidos para el brazo robotico Kawasaki RS03N

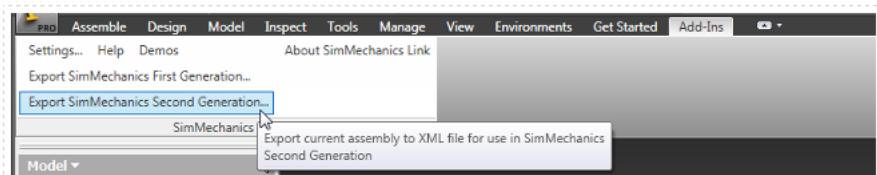
Cuadro 2: Parametros Denavit-Hartenberg

J	Theta	d	a	alpha
1	q1	0.168	0.119	$\pi/2$
2	q2	0	0.25	0
3	q3	0	0	$-\pi/2$
4	q4	0.223	0	$\pi/2$
5	q4	0	0	$\pi/2$
6	q5	0.07	0	0

Donde (theta) representa la rotación de la articulación, (d) corresponde al desfase o distancia a lo largo del eje de rotación hasta el próximo eje, (a) corresponde a la distancia perpendicular de un eje de rotación hasta el próximo, y (alpha) corresponde al ángulo existente entre un eje de rotación y el anterior.

Para la descarga del SimMechanics Link se debe ir a la siguiente pagina web [https://www.mathworks.com/products/simmechanics/download\\_smlink.html](https://www.mathworks.com/products/simmechanics/download_smlink.html), se debe seleccionar el paquete correspondiente a la versión de matlab instalada (en este caso 2012a), se procede a su instalación (las instrucciones se encuentran en la misma pagina de descarga); una vez instalado se procede a registrar el software CAD (en este caso Inventor 2013), escribiendo la siguiente instrucción `smlink_linkinv` en la linea de comando de matlab, finalmente matlab enviará un mensaje diciendo que el proceso ha sido satisfactorio. Posterior a esto se abre Inventor, y en la pestaña de complemento sera posible observar la opción de exportar a SimMechanics, como se muestra en la siguiente figura2.10.

Figura 2.10: Complemento SimMechanics en Inventor

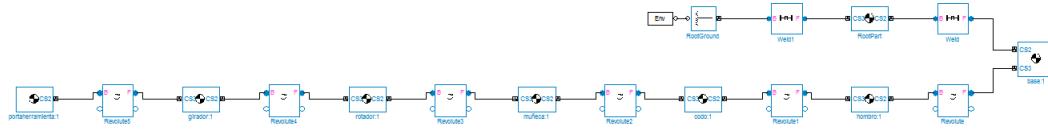


Con la ayuda del toolbox SimMechanics Link, se procede a convertir el modelo 3D de inventor a un formato comprensible por Matlab, en este caso la extensión de la conversión genera un archivo con extensión .xml que permitirá desde Matlab la importación de los parámetros físicos constructivos del robot.

En Inventor es necesario guardar una copia del modelo con la extensión .stl, que hará posible que la importación en matlab del entorno de simulación.

Finalmente dentro de matlab se corre el comando `mech_import(ruta de ubicación del archivo con extensión .xml)`, esto creará un diagrama de bloque como el que se muestra en la figura2.11.

Figura 2.11: Diagrama de bloques del ensamble



Finalmente se procede a guardar el diagrama en un archivo con extensión “mdl”, de esta manera ya es posible interactuar con el ensamble del robot. Los bloques creados contienen toda la información, de tipo constructivo como materiales, dimensiones, momentos de inercia y centros de gravedad, necesario para el análisis cinemático y dinámico del sistema. Este diagrama se encuentra con mas detalle en el anexo 1.

A partir de este punto, con la ayuda del Robotics Toolbox[1]se procede a realizar el análisis cinemático del robot, el procedimiento se describe a continuación.

De manera inicial se procede a la creación de los eslabones tomando como referencia los parámetros del cuadro2, en la figura2.12 se observa la introducción de estos valores (El ultimo valor de cada link corresponde al offset de correspondiente a cada theta de la articulación respectiva.

Figura 2.12: Creación de los eslabones

```
L1=Link([0 0.168 0.119 pi/2 0 0]);
L2=Link([0 0 0.250 0 0 pi/2]);
L3=Link([0 0 0 -pi/2 0 -pi/2]);
L4=Link([0 0.223 0 pi/2 0 0]);
L5=Link([0 0 0 pi/2 0 pi/2]);
L6=Link([0 0.07 0 0 0 0]);
```

Se extraen los valores de las masas de cada uno de los eslabones a partir de la construcción del modelo 3D con la ayuda de la siguiente instrucción (ver figura2.13).

Figura 2.13: Definición de las masas de los eslabones

```
L1.m=str2num(get_param('brazo/hombro:1','mass'));
L2.m=str2num(get_param('brazo/codo:1','mass'));
L3.m=str2num(get_param('brazo/muñeca:1','mass'));
L4.m=str2num(get_param('brazo/rotador:1','mass'));
L5.m=str2num(get_param('brazo/girador:1','mass'));
L6.m=str2num(get_param('brazo/portaherramienta:1','mass'));
```

Con la ayuda del siguiente algoritmo se extraen las inercias del modelo (ver figura2.14).

Figura 2.15: Centros de masa

```
L1.r=[23.8194 14.5372 164.384];
L2.r=[119.357 128.164 345.758];
L3.r=[119 15.3118 546.777];
L4.r=[119 -0.75 677.92];
L5.r=[140.15 -0.749706 747.039];
L6.r=[178.466 -0.75 747.039];
```

Figura 2.14: Inercias de los eslabones

```
L1.I=str2num(get_param('brazo/hombro:1','inertia'));
L2.I=str2num(get_param('brazo/codo:1','inertia'));
L3.I=str2num(get_param('brazo/muñeca:1','inertia'));
L4.I=str2num(get_param('brazo/rotador:1','inertia'));
L5.I=str2num(get_param('brazo/girador:1','inertia'));
L6.I=str2num(get_param('brazo/portaherramienta:1','inertia'));
```

Finalmente se extraen los centros de masa de cada uno de los eslabones con la ayuda del siguiente código, (ver figura2.15).

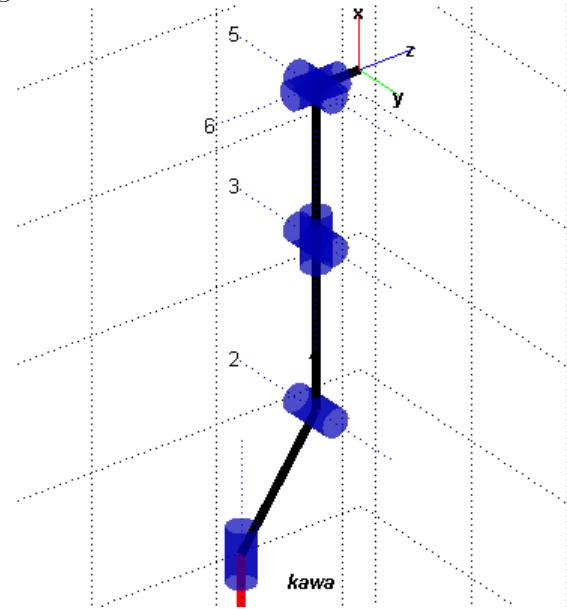
Ahora se procede a enlazar los eslabones y a mostrarlos para determinar que los datos previamente obtenidos son correctos, es claro que la posición de home definida para el robot, debe coincidir con la construcción asistida por el Robotics Toolbox, en la siguiente figura2.16 se muestra en código para lograrlo.

Figura 2.16: Enlace de los eslabones

```
%se enlazan los eslabones
R=SerialLink([L1,L2,L3,L4,L5,L6])
R.name=('kawa');
%se muestra grafico de los eslabones
plot(R,[0 0 0 0 0 0])
```

Cuando el código se corre se observa, (ver figura2.17), la unión de los eslabones como estructura de alambre, permitiendo definir que los valores han sido obtenidos de manera correcta.

Figura 2.17: Eslabones en estructura de alambre



Para continuar con el análisis se definen posiciones del brazo que permitirán realizar el análisis cinemático directo e inverso, los datos obtenidos en el análisis de cinemática directa permiten establecer la posición del efecto final con respecto al origen o base del robot.

En la figura 2.18 se muestran las posiciones definidas para el brazo, la posición  $p_0$  corresponde a la ubicación de home, por lo tanto sus valores corresponden todos a cero.

Figura 2.18: Posiciones definidas

```

p0=[0 0 0 0 0 0];
p1=[0 -pi/2 0 0 0 0 ];
p2=[0 -pi/2 0 0 pi/2 0];
p3=[0 -pi/2 -pi/4 0 3*pi/4 0];
p4=[pi/2 -pi/2 -pi/4 0 3*pi/4 0];

```

Con el comando *fkine* permite realizar el análisis de cinemática directa sobre el robot como se muestra en la siguiente figura 2.19. El análisis se realiza para cada una de las trayectorias propuestas en la figura 2.18.

Figura 2.19: Análisis cinemática directa

```
Cd0=R.fkine(p0)
Cd1=R.fkine(p1)
Cd2=R.fkine(p2)
Cd3=R.fkine(p3)
Cd4=R.fkine(p4)
```

En la siguiente figura 2.20 se muestra los resultados de cinemática directa para la posición p1, lo que permite establecer la posición final del portaherramienta.

Figura 2.20: Posición del efecto final

```
Cd1 =
1.0000      0      0      0.5920
-0.0000   -1.0000   -0.0000   -0.0000
0      0.0000   -1.0000      0.0980
0          0          0      1.0000
```

La cinemática inversa, permite determinar el movimiento necesario para lograr que el efecto final se ubique en una posición definida, es importante aclarar que la solución obtenida en este análisis no es única.

El análisis de cinemática inversa se logra con el uso de comando *ikine*, como se muestra en la siguiente figura.

Figura 2.21: Análisis cinemática inversa

```
NOAP1=Cd0;
NOAP2=Cd1;
NOAP3=Cd2;
NOAP4=Cd3;
NOAP5=Cd4;
M=[1 1 1 1 1 1]; %define los grados de libertad, 1 por cada uno
Q1=R.ikine(NOAP1,p0,M)
Q2=R.ikine(NOAP2,p1,M)
Q3=R.ikine(NOAP3,p2,M)
Q4=R.ikine(NOAP4,p3,M)
Q5=R.ikine(NOAP5,p4,M)
```

El resultado obtenido para la posición p1 permite observar que solamente la segunda articulación necesita giro de  $-\pi/2$  para llegar a la posición deseada, en la siguiente figura 2.22 se muestra el resultado obtenido.

$Q_2 =$

$$0 \quad -1.5708 \quad 0 \quad 0 \quad 0 \quad 0$$

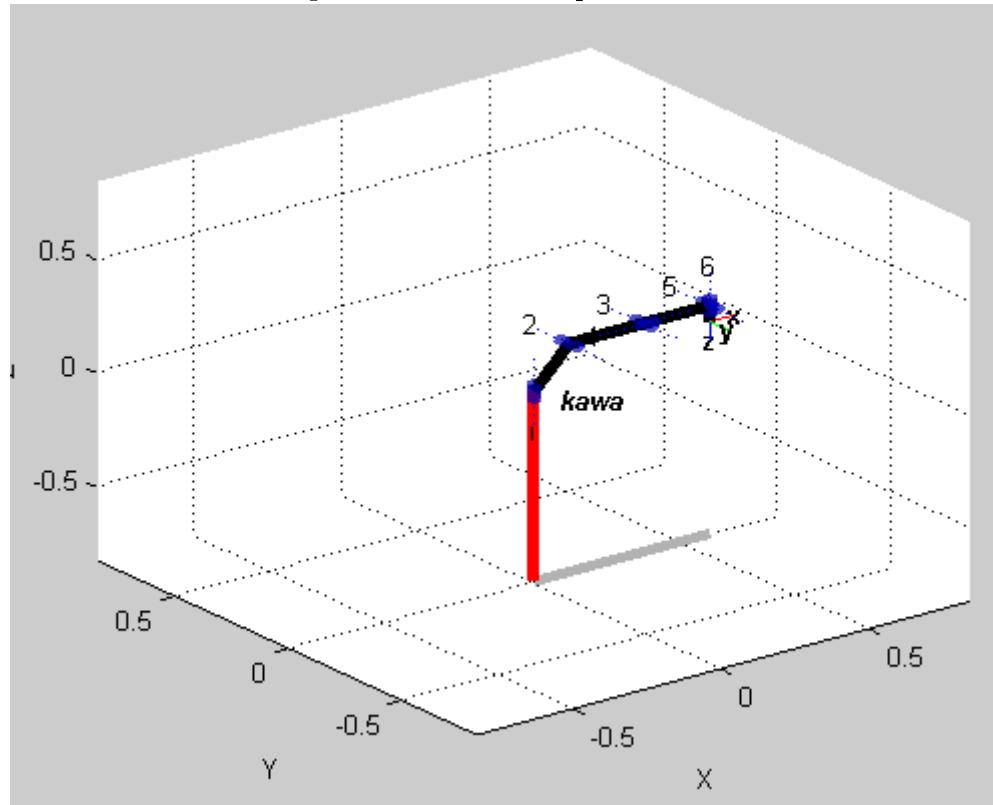
Finalmente es posible observar el movimiento del brazo sobre las trayectorias previamente establecidas con la ayuda del siguiente código, (ver figura 2.23).

Figura 2.23: Generación de trayectorias

```
[Q, QD, QDD]=jtraj(p0,p1,t);  
[Q1, QD1, QDD1]=jtraj(p1,p2,t);  
[Q2, QD2, QDD2]=jtraj(p2,p3,t);  
[Q3, QD3, QDD3]=jtraj(p3,p4,t);  
[Q4, QD4, QDD4]=jtraj(p4,p0,t);  
plot(R,[jtraj(p0,p1,t);jtraj(p1,p2,t);jtraj(p2,p3,t);jtraj(p3,p4,t);jtraj(p4,p0,t)])
```

En la siguiente figura 2.24, se observa el brazo en la posición p1 previamente establecida.

Figura 2.24: Posición p1 de brazo



### 3. DINÁMICA DEL BRAZO ROBÓTICO

El análisis dinámico permite determinar las fuerzas necesarias para lograr movimiento.

El análisis dinámico permitirá el análisis de torsión necesario para cada articulación y reproducir bajo el ambiente simulado el comportamiento del robot en condiciones de operación. Es claro que para el desarrollo del análisis dinámico, es necesario el análisis cinemático del sistema en conjunto con la completa definición de las características físicas del sistema asociados a las trayectorias definidas para el efecto final.

Dentro del análisis dinámico, existen la aproximación dinámica directa e inversa, y es partir de la segunda que se trabajará para poder determinar las fuerzas y momentos que intervienen en un sistema mecánico cuando el movimiento del mismo es conocido, como es el caso de este proyecto.

En el capítulo anterior se muestra como se generan las trayectorias, en el se usa el comando jtraj, que permite el estudio de una trayectoria del espacio articular entre dos puntos diferentes entre dos puntos diferentes, este comando realiza una curva de posición angular por cada paso analizado y permite también definir velocidad y aceleración de los puntos analizados.[3]

De manera inicial se usan las trayectorias previamente establecidas en la figura2.18, luego es necesario definir el vector de tiempo para el desarrollo de las trayectorias y finalmente se generan las trayectorias, como se muestra en la figura3.1.

Figura 3.1: Generación de trayectorias

```
%se define el vector tiempo para la trayectoria
t=(0:0.05:2);

%gereracion de trayectorias

[Q, QD, QDD]=jtraj(p0,p1,t);
[Q1, QD1, QDD1]=jtraj(p1,p2,t);
[Q2, QD2, QDD2]=jtraj(p2,p3,t);
[Q3, QD3, QDD3]=jtraj(p3,p4,t);
[Q4, QD4, QDD4]=jtraj(p4,p0,t);
plot(R,[jtraj(p0,p1,t);jtraj(p1,p2,t);jtraj(p2,p3,t);jtraj(p3,p4,t);jtraj(p4,p0,t)])
```

las variables Q, QD, QDD representan respectivamente los valores de posición, velocidad y aceleración de cada uno de los pares cinemáticos del sistema, lo que permite observar la respuesta de cada uno de los pares de manera independiente.

### 3.1. ANÁLISIS DINÁMICO INVERSO

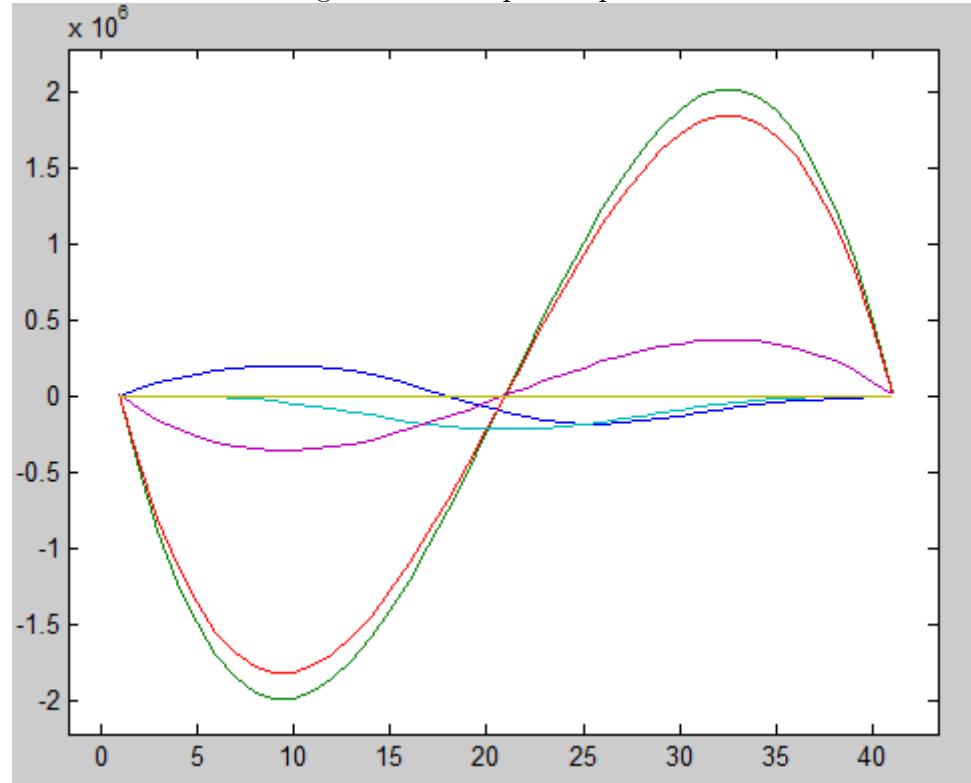
A partir de las trayectorias creadas es posible realizar el análisis que permita determinar el torque necesario a través de la función *rne* que mostrará los valores aplicados para lograr cada una de las posiciones establecidas como se muestra en la siguiente figura3.2.

Figura 3.2: análisis dinámico inverso

```
tau=R.rne(Q,QD,QDD);  
tau1=R.rne(Q1,QD1,QDD1);  
tau2=R.rne(Q2,QD2,QDD2);  
tau3=R.rne(Q3,QD3,QDD3);  
tau4=R.rne(Q4,QD4,QDD4);
```

En la siguiente figura3.3, se puede observar los torques requeridos para ir de la posición p0 a la posición p1, cada uno de los ejes aparece en un color específico.

Figura 3.3: Torques requeridos



En color verde se observa el torque requerido por el hombro del robot.  
en color rojo se observa el torque requerido por el codo del robot.

en color violeta se observa el torque requerido por la base del robot.

en color azul aparece el torque requerido por el girador del robot.

en color verde claro aparece el torque requerido por el efecto final.

en color amarillo aparece el torque requerido por el rotador del robot.

Se evidencia claramente que los torques mayores son requeridos en la articulaciones que mayor peso deben soportar por efecto de las características constructivas del mismo. Estas son el hombro y el codo.

## 4. INTERFAZ DE CONTROL DEL ROBOT

Para la construcción de la interfaz se usará como herramienta de apoyo, el robotics toolbox y se creará una gui en matlab que permita una fácil interacción con el robot.

Para continuar será necesario crear elementos adicionales en el diagrama de bloques que permita la interacción de estos elementos con el código de matlab. La primer parte consta de la creación de subsistemas (ver figura4.1), los cuales están conectados a cada uno de las articulaciones del robot; dentro de cada subsistema, se insertan tres bloques “from Workspace” que permiten la señales de control desde el código de matlab, un bloque para definir la posición, otro para definir la velocidad y otro para definir la aceleración de la articulación, estos van conectados a un “Joint Actuator” hace las veces de servomotor y transite movimiento a la articulación como se muestra en la figura4.2.

Figura 4.1: Conexión subsistemas

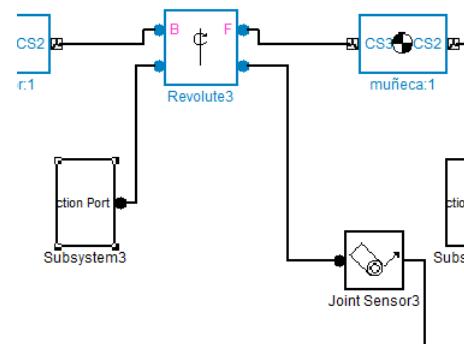
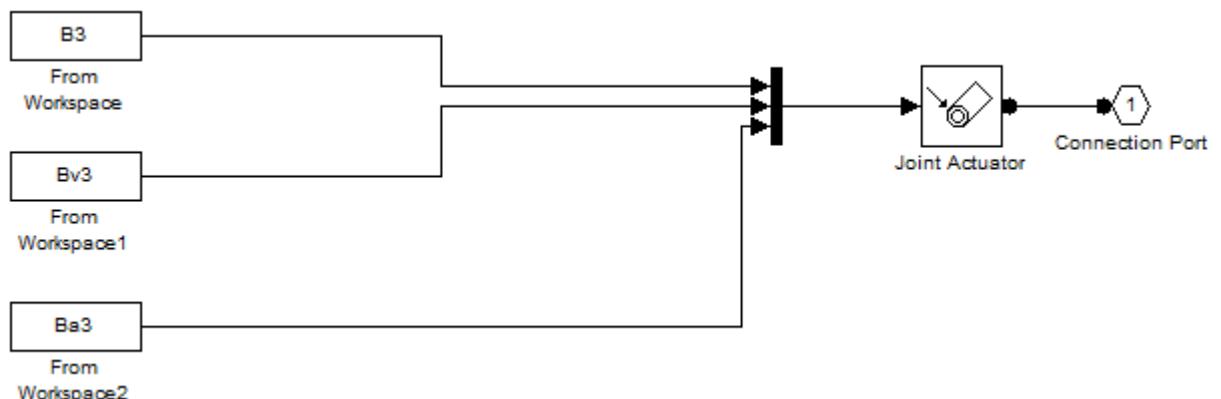


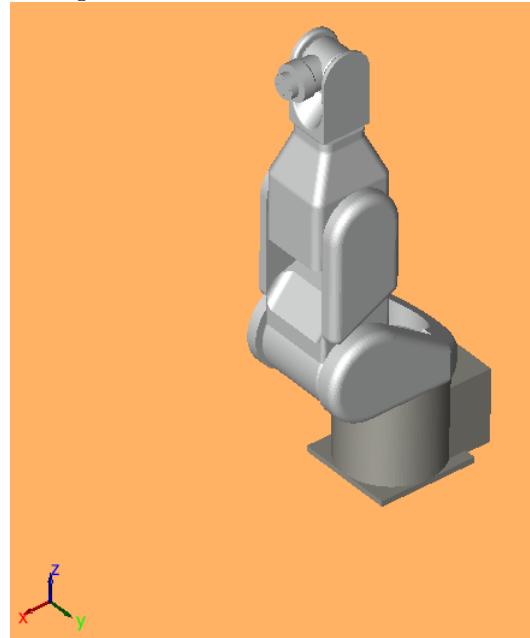
Figura 4.2: Detalle bloque subsistema



En el anexo 1, se puede ver en detalle el resultado final de la configuración del diagrama de bloques.

Una vez finalizada la configuración inicial, se procede a correr el sistema, para determinar que todo ha sido creado de manera satisfactoria, como resultado se obtiene un modelo en 3D, que contiene todos los parámetros constructivos del modelo, como se muestra en la siguiente figura4.3.

Figura 4.3: Modelo 3D en matlab



Finalmente se procede a crear una “gui” en matlab que permita la interacción de usuario con el robot. De manera inicial se propone una ventana de bienvenida, la cual dará acceso a la funciones del robot, como se muestra en la siguiente figura4.4, en ella se proponen dos simulaciones, la primera que permite el movimiento de cada una de las articulaciones, y la segunda que permite observar una simulación de movimiento del robot.

Figura 4.4: Pantalla de Inicio



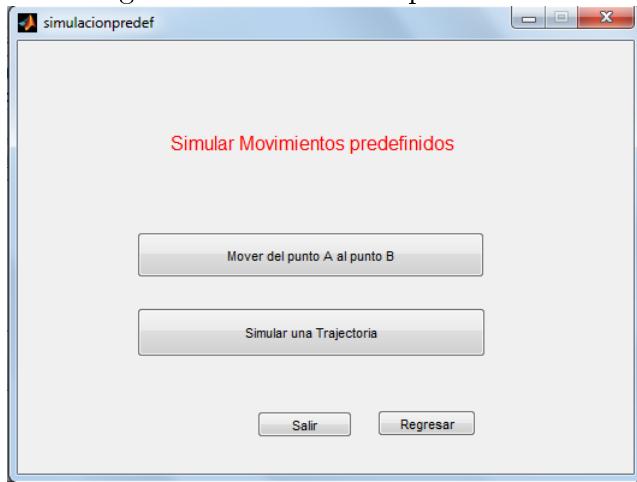
Cuando se selecciona la primer opción el sistema muestra una nueva pantalla, como la que se observa en la figura4.5, en ella hay unas casillas en las que se puede insertar el valor en grados que se quiere rotar cada una de las articulaciones, y un botón de inicio de simulación que permite observar el resultado, de igual forma hay un botón de retorno a la pantalla de inicio y uno para cerrar el entorno de simulación.

Figura 4.5: Movimiento de las articulaciones



El botón de simulación predefinida lleva a una pantalla como la que se muestra en la siguiente figura4.6, que permite ver dos movimientos, ir de un punto A a un punto B y la segunda opción el desarrollo de una trayectoria predefinida, y al igual que en la pantalla anterior la opción de retornar a la pantalla principal o salir de la simulación.

Figura 4.6: Simulación predefinida



Para terminar se procede a probar que la interfaz gráfica creada opere de manera correcta, permitiendo la interacción con el brazo robótico.

El código usado para la creación de la interfaz gráfica se puede observar con detalle en el anexo 2.

## 5. CONCLUSIONES Y RECOMENDACIONES

- Los programas de diseño asistido por computador son herramienta poderosa que permite de manera sencilla la construcción de mecanismos, teniendo en cuenta características geométricas y constructivas como el tipo de material, la determinación de ajustes y restricciones entre elementos del mecanismo, posibilitando realizar de manera rápida el análisis cinemático y dinámico del robot o cualquier sistema que se deseé evaluar. La aplicación de la ingeniería inversa, permite conocer de manera directa como están construidos los robots, y acortar la curva de aprendizaje en cuanto a la construcción y componentes que lo conforman. El software CAD, permitió de manera sencilla obtener los parámetros de centros de gravedad y momentos de inercia necesarios para los análisis de este proyecto, resultado que obtenidos de manera teórica podían tornarse complejos, dadas las características geométricas del robot. Por lo tanto se puede decir que la parte de diseño del robot se logra a cabalidad, logrando simulaciones estructurales y de movimiento.
- El análisis cinemático y dinámico con la ayuda de matlab y el robotics toolbox, ofrecen una alternativa viable y sencilla, ya que se integra de manera directa con el modelo creado en cualquier software CAD. La implementación del método Denavit-Hartenberg facilita la obtención de los parámetros cinemáticos, que posibilitan el análisis dinámico del robot y la determinación de los torques necesarios para que el sistema pueda realizar los movimientos requeridos. El toolbox ofrece una herramienta suficiente para el desarrollo del análisis del robot, se recomienda analizar más a fondo sus funciones y herramientas.
- La utilización del SimMechanics, ofrece un entorno de trabajo claro y amigable al usuario, a través de él se pudo comparar que los datos como masas, momentos de inercia y centros de gravedad coinciden con los creados en el modelo del software CAD, por lo tanto se recomienda un análisis más exhaustivo de esta herramienta ya que permite la simulación e implementación de algoritmos de control sobre modelos virtuales, con características similares pero no decir idénticas a los modelos reales.
- Este documento permite un acercamiento inicial al desarrollo de robots manipuladores, se evidencia que cada programa usado es fuerte en su área específica, ya que Inventor permite la creación de modelos apegados a la realidad, cosa que Matlab por si no permite hacer fácilmente, y Matlab permite el modelamiento y análisis matemático, de manera clara y sencilla, algo que ya es muy bien sabido por la comunidad educativa. Es necesario seguir con el estudio de estas herramientas, en pro de poder iniciar un desarrollo tecnológico que pueda impactar de manera positiva el sector industrial del país.
- El desarrollo de la interfaz gráfica permite comprender de manera directa los criterios básicos necesarios para la programación de trayectorias en un dispositivo

robótico, el programa aquí creado es una etapa inicial que puede ser mejorada agregando la interacción con diferentes estrategias de control del manipulador y otros análisis que pueden resultar relevantes a la hora de explorar la cinemática y la dinámica de este tipo de dispositivos.

## Referencias

- [1] P.I. Corke. *Robotics, Vision & Control*. Springer, 2011.
- [2] Instituto Nacional de Tecnologias Educativas y de Formacion del Profesorado. Robots industriales. Electronica, noviembre 2013.
- [3] S. D. Vaca Jimenez. Diseno y simulacion de un robot manipulador industrial tipo puma capaz de levantar pesos de hasta 20 kg. Technical report, Escuela politecnica nacional, 2012.
- [4] J. F. Reyes Cortes. *Robotica control de robots manipuladores*. Alfaomega Grupo editor, S.A. de C.V., 2001.
- [5] Sergio Saldana Sanchez. Mecanismos. Electronica, Noviembre 2013.

## A. ANEXO 1: DIAGRAMA DE BLOQUES EN SIMULINK

Figura A.1: Diagrama de bloques inicial Simulink

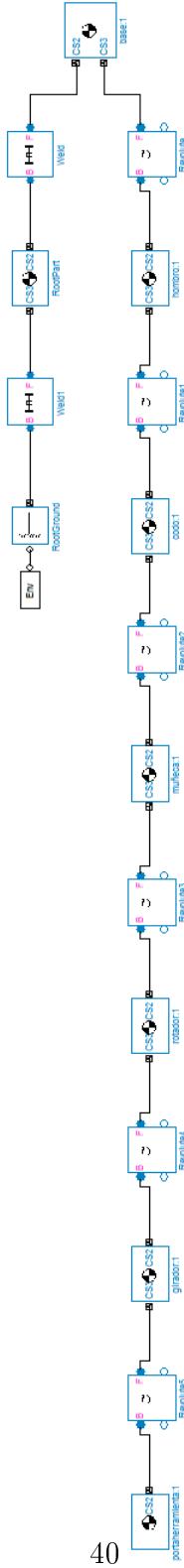
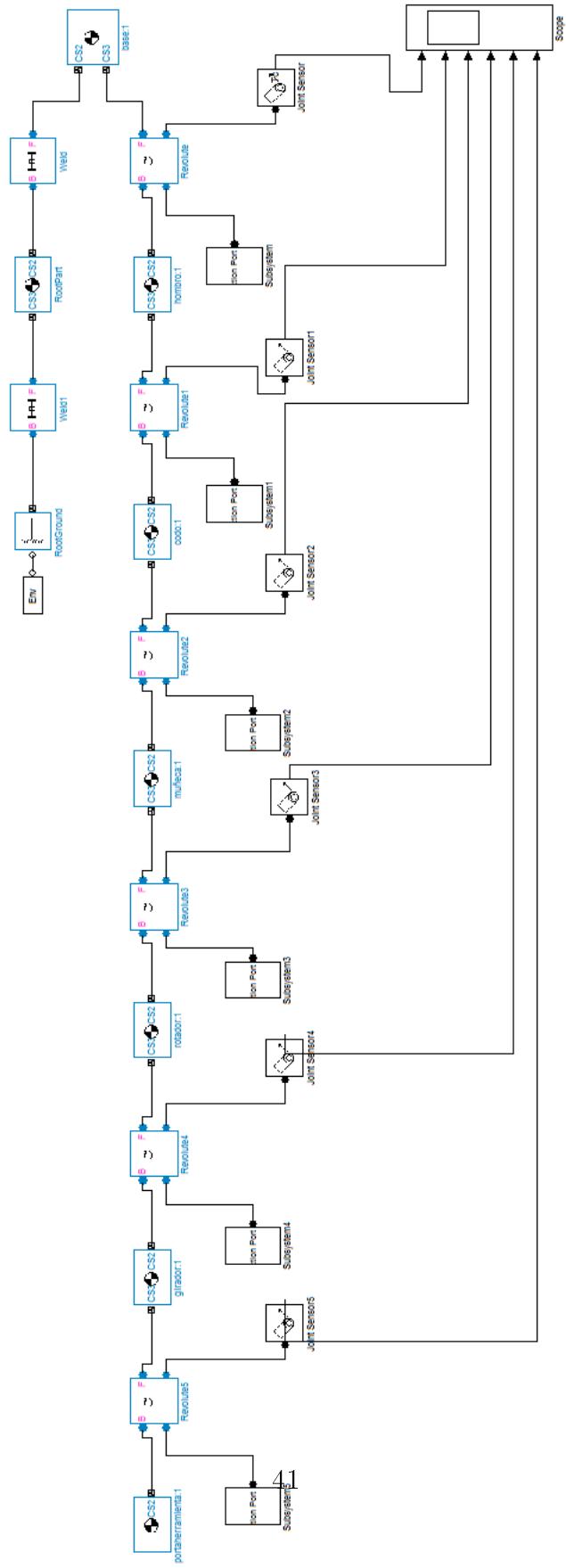


Figura A.2: Diagrama de bloques final Simulink



## B. ANEXO 2: CÓDIGO DE DESARROLLO DE LA INTERFAZ GRÁFICA

Figura B.1: Código matlab

```
function varargout = kawa(varargin)

gui_Singleton = 1;
gui_State = struct('gui_Name',        mfilename, ...
                   'gui_Singleton',   gui_Singleton, ...
                   'gui_OpeningFcn', @kawa_OpeningFcn, ...
                   'gui_OutputFcn',  @kawa_OutputFcn, ...
                   'gui_LayoutFcn',  [], ...
                   'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before kawa is made visible.
function kawa_OpeningFcn(hObject, eventdata, handles, varargin)

handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

function varargout = kawa_OutputFcn(hObject, eventdata, handles)

varargout{1} = handles.output;

% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)

close
run simulacionlibre

% --- Executes on button press in pushbutton2.
function pushbutton2_Callback(hObject, eventdata, handles)

close
run simulacionpredef

% --- Executes on button press in pushbutton3.
function pushbutton3_Callback(hObject, eventdata, handles)

close
```

Figura B.2: Simulación libre

```
startup_rvc
open_system('brazo.mdl')
L1=Link([0 0.168 0.119 pi/2 0 0]);% se crean los elabones a partir de
L2=Link([0 0 0.250 0 0 pi/2]);      %parametros Denavit-Hartenberg
L3=Link([0 0 0 -pi/2 0 -pi/2]);
L4=Link([0 0.223 0 pi/2 0 0]);
L5=Link([0 0 0 pi/2 0 pi/2]);
L6=Link([0 0.07 0 0 0 0]);

L1.m=str2num(get_param('brazo/hombro:1','mass'));
L2.m=str2num(get_param('brazo/codo:1','mass'));
L3.m=str2num(get_param('brazo/muñeca:1','mass'));
L4.m=str2num(get_param('brazo/rotador:1','mass'));
L5.m=str2num(get_param('brazo/girador:1','mass'));
L6.m=str2num(get_param('brazo/portaherramienta:1','mass'));

L1.I=str2num(get_param('brazo/hombro:1','inertia'));
L2.I=str2num(get_param('brazo/codo:1','inertia'));
L3.I=str2num(get_param('brazo/muñeca:1','inertia'));
L4.I=str2num(get_param('brazo/rotador:1','inertia'));
L5.I=str2num(get_param('brazo/girador:1','inertia'));
L6.I=str2num(get_param('brazo/portaherramienta:1','inertia'));

L1.r=[23.8194 14.5372 164.384];
L2.r=[119.357 128.164 345.758];
L3.r=[119 15.3118 546.777];
L4.r=[119 -0.75 677.92];
L5.r=[140.15 -0.749706 747.039];
L6.r=[178.466 -0.75 747.039];

R=SerialLink([L1,L2,L3,L4,L5,L6]);
R.name=('kawa');
A=imread('ejes','bmp');
Img=image(A,'Parent',handles.axes2);
set(handles.axes2,'Visible','off','Ydir','reverse','Xlim',...
    ...)
```

Figura B.3: Simulación libre

```
01=deg2rad(str2double(get(handles.edit1,'string')));
02=deg2rad(str2double(get(handles.edit2,'string')));
03=deg2rad(str2double(get(handles.edit3,'string')));
04=deg2rad(str2double(get(handles.edit4,'string')));
05=deg2rad(str2double(get(handles.edit5,'string')));
06=deg2rad(str2double(get(handles.edit6,'string')));
p0=[0 0 0 0 0];
p1=[01 02 03 04 05 06];
t=(0:0.02:400);
[q,qd,qdd]=jtraj(p0,p1,t);

B1(:,1)=(t);
B1(:,2)=q(:,1);
assignin('base','B1',B1)
Bv1(:,1)=(t);
Bv1(:,2)=qd(:,1);
assignin('base','Bv1',Bv1);
Ba1(:,1)=(t);
Ba1(:,2)=qdd(:,1);
assignin('base','Ba1',Ba1);

B2(:,1)=(t);
B2(:,2)=q(:,2);
assignin('base','B2',B2)
Bv2(:,1)=(t);
Bv2(:,2)=qd(:,2);
assignin('base','Bv2',Bv2);
Ba2(:,1)=(t);
Ba2(:,2)=qdd(:,2);
assignin('base','Ba2',Ba2);
```

Figura B.4: Simulación libre

```
B3(:,1)=(t);
B3(:,2)=q(:,3);
assignin('base','B3',B3)
Bv3(:,1)=(t);
Bv3(:,2)=qd(:,3);
assignin('base','Bv3',Bv3);
Ba3(:,1)=(t);
Ba3(:,2)=qdd(:,3);
assignin('base','Ba3',Ba3);

B4(:,1)=(t);
B4(:,2)=q(:,4);
assignin('base','B4',B4)
Bv4(:,1)=(t);
Bv4(:,2)=qd(:,4);
assignin('base','Bv4',Bv4);
Ba4(:,1)=(t);
Ba4(:,2)=qdd(:,4);
assignin('base','Ba4',Ba4);

B5(:,1)=(t);
B5(:,2)=q(:,5);
assignin('base','B5',B5)
Bv5(:,1)=(t);
Bv5(:,2)=qd(:,5);
assignin('base','Bv5',Bv5);
Ba5(:,1)=(t);
Ba5(:,2)=qdd(:,5);
assignin('base','Ba5',Ba5);
```

Figura B.5: Simulación libre

```
B6(:,1)=(t);
B6(:,2)=q(:,6);
assignin('base','B6',B6)
Bv6(:,1)=(t);
Bv6(:,2)=qd(:,6);
assignin('base','Bv6',Bv6);
Ba6(:,1)=(t);
Ba6(:,2)=qdd(:,6);
assignin('base','Ba6',Ba6);
sim('brazo.mdl')
```

Figura B.6: Simulación predefinida

```
clear
P1=deg2rad([0 0 0 0 0 0]);
P2=deg2rad([40 64.8 86.4 0 -61.2 0]);

t=0:0.02:400;

[q,qd,qdd]=jtraj(P1,P2,t);

B1(:,1)=(t);
B1(:,2)=q(:,1);
assignin('base','B1',B1)
Bv1(:,1)=(t);
Bv1(:,2)=qd(:,1);
assignin('base','Bv1',Bv1);
Ba1(:,1)=(t);
Ba1(:,2)=qdd(:,1);
assignin('base','Ba1',Ba1);

B2(:,1)=(t);
B2(:,2)=q(:,2);
assignin('base','B2',B2)
Bv2(:,1)=(t);
Bv2(:,2)=qd(:,2);
assignin('base','Bv2',Bv2);
Ba2(:,1)=(t);
Ba2(:,2)=qdd(:,2);
assignin('base','Ba2',Ba2);

B3(:,1)=(t);
B3(:,2)=q(:,3);
assignin('base','B3',B3)
Bv3(:,1)=(t);
Bv3(:,2)=qd(:,3);
assignin('base','Bv3',Bv3);
Ba3(:,1)=(t);
Ba3(:,2)=qdd(:,3);
assignin('base','Ba3',Ba3);
```

Figura B.7: Simulación predefinida

```
B4(:,1)=(t);
B4(:,2)=q(:,4);
assignin('base','B4',B4)
Bv4(:,1)=(t);
Bv4(:,2)=qd(:,4);
assignin('base','Bv4',Bv4);
Ba4(:,1)=(t);
Ba4(:,2)=qdd(:,4);
assignin('base','Ba4',Ba4);

B5(:,1)=(t);
B5(:,2)=q(:,5);
assignin('base','B5',B5)
Bv5(:,1)=(t);
Bv5(:,2)=qd(:,5);
assignin('base','Bv5',Bv5);
Ba5(:,1)=(t);
Ba5(:,2)=qdd(:,5);
assignin('base','Ba5',Ba5);

B6(:,1)=(t);
B6(:,2)=q(:,6);
assignin('base','B6',B6)
Bv6(:,1)=(t);
Bv6(:,2)=qd(:,6);
assignin('base','Bv6',Bv6);
Ba6(:,1)=(t);
Ba6(:,2)=qdd(:,6);
assignin('base','Ba6',Ba6);
sim('brazo.mdl')
pause(0.1)
P1=P2;
P2=deg2rad([-40 64.8 68.4 0 -61.2 0]);

[q,qd,qdd]=jtraj(P1,P2,t);
```

Figura B.8: Simulación predefinida

```
B1(:,1)=(t);
B1(:,2)=q(:,1);
assignin('base','B1',B1)
Bv1(:,1)=(t);
Bv1(:,2)=qd(:,1);
assignin('base','Bv1',Bv1);
Ba1(:,1)=(t);
Ba1(:,2)=qdd(:,1);
assignin('base','Ba1',Ba1);

B2(:,1)=(t);
B2(:,2)=q(:,2);
assignin('base','B2',B2)
Bv2(:,1)=(t);
Bv2(:,2)=qd(:,2);
assignin('base','Bv2',Bv2);
Ba2(:,1)=(t);
Ba2(:,2)=qdd(:,2);
assignin('base','Ba2',Ba2);

B3(:,1)=(t);
B3(:,2)=q(:,3);
assignin('base','B3',B3)
Bv3(:,1)=(t);
Bv3(:,2)=qd(:,3);
assignin('base','Bv3',Bv3);
Ba3(:,1)=(t);
Ba3(:,2)=qdd(:,3);
assignin('base','Ba3',Ba3);

B4(:,1)=(t);
B4(:,2)=q(:,4);
assignin('base','B4',B4)
Bv4(:,1)=(t);
Bv4(:,2)=qd(:,4);
assignin('base','Bv4',Bv4);
Ba4(:,1)=(t);
Ba4(:,2)=qdd(:,4);
assignin('base','Ba4',Ba4); 49
```

Figura B.9: Simulación predefinida

```
B5(:,1)=(t);
B5(:,2)=q(:,5);
assignin('base','B5',B5)
Bv5(:,1)=(t);
Bv5(:,2)=qd(:,5);
assignin('base','Bv5',Bv5);
Ba5(:,1)=(t);
Ba5(:,2)=qdd(:,5);
assignin('base','Ba5',Ba5);

B6(:,1)=(t);
B6(:,2)=q(:,6);
assignin('base','B6',B6)
Bv6(:,1)=(t);
Bv6(:,2)=qd(:,6);
assignin('base','Bv6',Bv6);
Ba6(:,1)=(t);
Ba6(:,2)=qdd(:,6);
assignin('base','Ba6',Ba6);
sim('brazo.mdl')
pause(0.1)
P1=P2;
P2=deg2rad([0 3.6 126 0 -39.6 0]);

[q,qd,qdd]=jtraj(P1,P2,t);

B1(:,1)=(t);
B1(:,2)=q(:,1);
assignin('base','B1',B1)
Bv1(:,1)=(t);
Bv1(:,2)=qd(:,1);
assignin('base','Bv1',Bv1);
Ba1(:,1)=(t);
Ba1(:,2)=qdd(:,1);
assignin('base','Ba1',Ba1);
```

Figura B.10: Simulación predefinida

```
B2(:,1)=(t);
B2(:,2)=q(:,2);
assignin('base','B2',B2)
Bv2(:,1)=(t);
Bv2(:,2)=qd(:,2);
assignin('base','Bv2',Bv2);
Ba2(:,1)=(t);
Ba2(:,2)=qdd(:,2);
assignin('base','Ba2',Ba2);

B3(:,1)=(t);
B3(:,2)=q(:,3);
assignin('base','B3',B3)
Bv3(:,1)=(t);
Bv3(:,2)=qd(:,3);
assignin('base','Bv3',Bv3);
Ba3(:,1)=(t);
Ba3(:,2)=qdd(:,3);
assignin('base','Ba3',Ba3);

B4(:,1)=(t);
B4(:,2)=q(:,4);
assignin('base','B4',B4)
Bv4(:,1)=(t);
Bv4(:,2)=qd(:,4);
assignin('base','Bv4',Bv4);
Ba4(:,1)=(t);
Ba4(:,2)=qdd(:,4);
assignin('base','Ba4',Ba4);

B5(:,1)=(t);
B5(:,2)=q(:,5);
assignin('base','B5',B5)
Bv5(:,1)=(t);
Bv5(:,2)=qd(:,5);
assignin('base','Bv5',Bv5);
Ba5(:,1)=(t);
Ba5(:,2)=qdd(:,5);
assignin('base','Ba5',Ba5); 51
```

Figura B.11: Simulación predefinida

```
B6(:,1)=(t);
B6(:,2)=q(:,6);
assignin('base','B6',B6)
Bv6(:,1)=(t);
Bv6(:,2)=qd(:,6);
assignin('base','Bv6',Bv6);
Ba6(:,1)=(t);
Ba6(:,2)=qdd(:,6);
assignin('base','Ba6',Ba6);
sim('brazo.mdl')
pause(0.1)
P1=P2
P2=deg2rad([0 0 0 0 0 0]);
[q,qd,qdd]=jtraj(P1,P2,t);

B1(:,1)=(t);
B1(:,2)=q(:,1);
assignin('base','B1',B1)
Bv1(:,1)=(t);
Bv1(:,2)=qd(:,1);
assignin('base','Bv1',Bv1);
Ba1(:,1)=(t);
Ba1(:,2)=qdd(:,1);
assignin('base','Ba1',Ba1);

B2(:,1)=(t);
B2(:,2)=q(:,2);
assignin('base','B2',B2)
Bv2(:,1)=(t);
Bv2(:,2)=qd(:,2);
assignin('base','Bv2',Bv2);
Ba2(:,1)=(t);
Ba2(:,2)=qdd(:,2);
assignin('base','Ba2',Ba2);
```

Figura B.12: Simulación predefinida

```
B3(:,1)=(t);
B3(:,2)=q(:,3);
assignin('base','B3',B3)
Bv3(:,1)=(t);
Bv3(:,2)=qd(:,3);
assignin('base','Bv3',Bv3);
Ba3(:,1)=(t);
Ba3(:,2)=qdd(:,3);
assignin('base','Ba3',Ba3);

B4(:,1)=(t);
B4(:,2)=q(:,4);
assignin('base','B4',B4)
Bv4(:,1)=(t);
Bv4(:,2)=qd(:,4);
assignin('base','Bv4',Bv4);
Ba4(:,1)=(t);
Ba4(:,2)=qdd(:,4);
assignin('base','Ba4',Ba4);

B5(:,1)=(t);
B5(:,2)=q(:,5);
assignin('base','B5',B5)
Bv5(:,1)=(t);
Bv5(:,2)=qd(:,5);
assignin('base','Bv5',Bv5);
Ba5(:,1)=(t);
Ba5(:,2)=qdd(:,5);
assignin('base','Ba5',Ba5);

B6(:,1)=(t);
B6(:,2)=q(:,6);
assignin('base','B6',B6)
Bv6(:,1)=(t);
Bv6(:,2)=qd(:,6);
assignin('base','Bv6',Bv6);
Ba6(:,1)=(t);
Ba6(:,2)=qdd(:,6);
assignin('base','Ba6',Ba6);    53
sim('brazo.mdl')
```

**C. ANEXO3: ARCHIVOS DE CONSTRUCCIÓN EN  
INVENTOR 2013**