

República Bolivariana de Venezuela
Universidad de Los Andes
Facultad de Ingeniería
Escuela de Sistemas
Especialización y Maestría en Ingeniería de Control y Automatización

DISEÑO DE UN MANIPULADOR ROBÓTICO CON TRES GRADOS DE LIBERTAD
PARA FINES EDUCATIVOS

WWW.BDIGITAL.ULA.VE

PROYECTO DE GRADO PRESENTADO ANTE LA ILUSTRE UNIVERSIDAD DE LOS
ANDES COMO REQUISITO FINAL PARA OPTAR AL TÍTULO DE MAGISTER
SCIENTIAE EN INGENIERÍA DE CONTROL Y AUTOMATIZACIÓN

Autor: Ing. Luis José Andueza Castro

Tutor: Dr. Iñaki Aguirre

Mérida, Abril de 2008

Licencia Creative Commons:
Atribución - No Comercial - Compartir Igual 3.0 Venezuela
(CC BY-NC-SA 3.0 VE)

Dedicatoria

A mi esposa Cároly, gracias por apoyarme siempre. Para ti con todo mi amor.

WWW.BDIGITAL.ULA.VE

Licencia Creative Commons:
Atribución - No Comercial - Compartir Igual 3.0 Venezuela
(CC BY-NC-SA 3.0 VE)

Agradecimientos

A Dios, por iluminarme el camino

A mi tutor Dr. Iñaki Aguirre, por su ayuda y paciencia

Este trabajo ha sido realizado con el apoyo recibido del C.D.C.H.T. de la Universidad de Los Andes, bajo el marco del proyecto I-919-06-02-B titulado Diseño y Construcción de un Manipulador Robótico para fines educativos.

WWW.BDIGITAL.ULA.VE

Licencia Creative Commons:
Atribución - No Comercial - Compartir Igual 3.0 Venezuela
(CC BY-NC-SA 3.0 VE)

Tabla de contenido

Diseño de un manipulador robótico con tres grados de libertad para fines educativos	i
Dedicatoria	ii
Agradecimientos.....	iii
Índice de Tablas	vi
Índice de Figuras	vii
Resumen	xii
Abstract	xii
Capítulo 1: Introducción	1
1.1 Planteamiento del problema	1
1.2 Justificación del proyecto	3
Capítulo 2: Estado del arte	6
2.1 Definición de robots y robots manipuladores.....	6
2.1.1 Clasificación de los robots manipuladores	7
2.1.2 Clasificación de los manipuladores según la configuración cinemática.	13
2.2 Investigación en robótica de manipuladores Industriales.....	15
Capítulo 3: Generalidades de los manipuladores	20
3.1 Herramientas matemáticas necesarias en el uso de manipuladores.....	20
3.2 Herramientas computacionales para el diseño de robots manipuladores	51
Capítulo 4: Brazo Manipulador propuesto (Prototipo)	55
4.1 Diseño Mecánico	56
4.2 Cinemática del manipulador propuesto	88
4.3 Comparación de cálculos.....	101

4.4 Herramienta computacional para la simulación y programación del manipulador robótico	118
4.5 Costos del robot manipulador propuesto	126
Capítulo 5: Conclusiones, recomendaciones y futuros trabajos	128
5.1 Conclusiones.....	128
5.2 Recomendaciones	129
5.3 Futuros trabajos	130
Bibliografía	131
Anexo A	133
Anexo B.....	150
Anexo C	167
Anexo D	169
Anexo E.....	178
Anexo F.....	180
Anexo G	183
Anexo H.....	198
Anexo I.....	205
Anexo J	210
Anexo K.....	215
Anexo L.....	220
Anexo M.....	240

Índice de Tablas

Tabla 1 Clasificación de las aplicaciones de los robots industriales manipuladores según IFR. ...	7
Tabla 2. Robots Manipuladores Comerciales, dimensiones en mm.	57
Tabla 3 Propuestas de tamaño para la propuesta de manipulador. Dimensiones en mm.	57
Tabla 4 Principales ventajas y desventajas de los distintos tipos de accionadores.....	68
Tabla 5 Motores paso a paso para la selección	70
Tabla 6 Motores paso a paso conseguidos nacionalmente.....	71
Tabla 7 Motor LEGO ® NXT	71
Tabla 8 Comparación entre distintos sensores de posición angular.	73
Tabla 9 Sistemas de transmisión.....	74
Tabla 10 Parámetros Denavit-Hartenberg para el manipulador robótico en estudio.....	89
Tabla 11 Peso de cada uno de los eslabones.....	101
Tabla 12 Comparación de resultados para la cinemática directa.....	102
Tabla 13 Comparación de las cinemáticas inversas.....	104
Tabla 14 Comparación de las matrices Jacobianas directas	105
Tabla 15 Comparación resultados Jacobianas inversas	106
Tabla 16 Tabla comparativa de simuladores.	119
Tabla 17 Costos de fabricación del robot manipulador	126
Tabla 18 Comparación de precios de robots manipuladores	126

Índice de Figuras

Figura 1. Manipulador para el paletizado.	8
Figura 2 Robot manipulador de materiales.	9
Figura 3 Manipulador para soldadura.	10
Figura 4 Manipulador para pintura de automóviles.	10
Figura 5 Manipulador para prototipado rápido.	11
Figura 6 Manipuladores de corte por chorro de agua.	12
Figura 7 Robocoaster, manipulador utilizado para el entretenimiento.	12
Figura 8 Robot cartesiano.	13
Figura 9 Robor Cilíndrico.	13
Figura 10 Robot Esférico.	14
Figura 11 Robot Scara.	14
Figura 12 Robot Articulado.	15
Figura 13 Manipulador Stanford.	16
Figura 14 Manipulador robótico PUMA.	16
Figura 15 Sistema cartesiano 2D	22
Figura 16 Sistema cartesiano 3D	22
Figura 17 Coordenadas polares.	23
Figura 18 Coordenadas cilíndricas.	24
Figura 19 Coordenadas esféricas	24
Figura 20 Sistemas de coordenadas principal OXYZ y móvil OUVW	26
Figura 21 Sistema coordenado girado sobre el eje X	27
Figura 22 Sistema coordenado girado sobre eje Y	28

Figura 23 Sistema coordinado girado sobre eje Z.....	28
Figura 24 Parámetros Denavit-Hartenberg para eslabón giratorio.	38
Figura 25 MATLAB 7.	52
Figura 26 Maple 11.....	53
Figura 27 Autodesk Inventor 2008.	54
Figura 28 Eslabones de los manipuladores.....	56
Figura 29 Medidas generales del manipulador	58
Figura 30 Primera propuesta, perfil en "U"	59
Figura 31 Segunda propuestas, perfil cuadrado.....	60
Figura 32 Eslabón número uno	61
Figura 33 Extensiones.....	62
Figura 34 Eslabón número dos	63
Figura 35 Eslabón número tres	63
Figura 36 Unión eslabones uno y dos	64
Figura 37 Unión eslabones dos y tres	64
Figura 38 Tapas superior e inferior de la base.....	65
Figura 39 Base del Manipulador.....	65
Figura 40 Unión de la base con el eslabón número uno	66
Figura 41 Movimientos eslabón uno. Vista superior	67
Figura 42 Movimientos eslabón dos.....	67
Figura 43 Movimientos eslabón tres.....	67
Figura 44 Motor LEGO NXT.	72
Figura 45 Motor LEGO NXT - Tren de engranes de reducción.....	72

Figura 46 Engranajes de LEGO seleccionados.....	75
Figura 47 Alineación de los engranajes	76
Figura 48 Columnas Technic ® de LEGO ®	76
Figura 49 Columnas LEGO #9 para la unión de los eslabones dos y tres.....	77
Figura 50 Unión de las columnas LEGO con el eslabón tres	77
Figura 51 Unión columnas LEGO #5 y #7 al eslabón dos	78
Figura 52 Elementos de unión LEGO utilizados	79
Figura 53 Ensamblaje de motor para eslabones.....	79
Figura 54 Motor ensamblado para eslabones.....	80
Figura 55 Ensamblaje del motor con eslabón dos	80
Figura 56 Motor ensamblado en eslabón dos	81
Figura 57 Motores ensamblados al eslabón uno	81
Figura 58 Alineación engranajes con motor de la base	82
Figura 59 Agujeros fondo de la base	83
Figura 60 Columnas LEGO para sostener el motor en la base	84
Figura 61 Columna #5 en la tapa superior de la base	84
Figura 62 Despiece motor para la base	85
Figura 63 Motor de la base ensamblado	85
Figura 64 Base ensamblada	86
Figura 65 Manipulador ensamblado	86
Figura 66 Prototipo del robot manipulador propuesto	87
Figura 67 Prototipo del robot manipulador propuesto.....	87
Figura 68 Identificación de los eslabones y las articulaciones	88

Figura 69 Sistemas coordinados de las articulaciones según Denavit-Hartenberg.....	89
Figura 70 Resultados de la trayectoria 01 para la cinemática a partir de las matrices homogéneas.	108
Figura 71 Resultados de la trayectoria 01 para la cinemática calculada geoméricamente.	109
Figura 72 Resultados de la trayectoria 02 para la cinemática a partir de las matrices homogéneas.	110
Figura 73 Resultados de la trayectoria 02 para la cinemática calculada geoméricamente.	110
Figura 74 Resultados de la trayectoria 03 para la cinemática a partir de las matrices homogéneas.	111
Figura 75 Resultados de la trayectoria 02 para la cinemática calculada geoméricamente.	112
Figura 76 Resultados obtenidos con el "Robotic Toolbox for MATLAB"	113
Figura 77 Torques obtenidos para la trayectoria 1.....	114
Figura 78 Torques obtenidos para la trayectoria 2.....	115
Figura 79 Torque obtenidos para la trayectoria 3.....	116
Figura 80 Comparación de los Torques obtenidos con el "Robotics toolbox for MATLAB" y las ecuaciones de Newton-Euler, para la trayectoria 1.....	117
Figura 81 Comparación de los Torques obtenidos con el "Robotics toolbox for MATLAB" y las ecuaciones de Newton-Euler, para la trayectoria 2.....	117
Figura 82 Comparación de los Torques obtenidos con el "Robotics toolbox for MATLAB" y las ecuaciones de Newton-Euler, para la trayectoria 3.....	118
Figura 83 Modelo 3D simplificado del manipulador robótico	122
Figura 84 Interfaz de la herramienta computacional.	122
Figura 85 Panel de la cinemática directa.	123

Figura 86 Panel de la cinemática inversa.....	123
Figura 87 Panel de Pasos.	124
Figura 88 Panel gráfico.....	125

WWW.BDIGITAL.ULA.VE

Licencia Creative Commons:
Atribución - No Comercial - Compartir Igual 3.0 Venezuela
(CC BY-NC-SA 3.0 VE)

Resumen

El presente trabajo describe los aspectos principales del modelado y diseño de un manipulador robótico de tres grados de libertad. El manipulador diseñado tendrá fines didácticos y será utilizado en las áreas de automatización industrial, robótica, y control de procesos como herramienta de apoyo didáctico en Ingeniería de Sistemas de la Universidad de Los Andes. Se calcula la cinemática directa e inversa haciendo uso de los parámetros de Denavit-Hartenberg. Adicionalmente se calcula la dinámica del manipulador. Se utilizan encoders como sensores de posición y como actuadores se utilizan servo-motores. El sistema de transmisión de potencia hacia los eslabones se realiza por medio de engranajes. Finalmente se presenta un prototipo de diseño del manipulador robótico adaptado a las necesidades educativas.

Palabras claves: Robótica, manipulador robótico, cinemática de manipuladores, dinámica de manipuladores.

Abstract

The following work describes the principal modelling and design aspects of a three degree of freedom robotic manipulator. The design manipulator will be for educational purposes in the areas of industrial automation, robotics, and process control in System Engineering at ULA. For the design the forward and inverse kinematics are calculated with the Denavit-Hartenberg parameters. The manipulator dynamics is also calculated. Sensors with angular position measurement are used. As an actuators servo-motors are used. The power transmission system to the links is done by gears. Finally a robotic manipulator design proposal is presented adapted to educational needs.

Key words: Robotics, Robotic manipulator, kinematics of manipulators, dynamics of manipulators

Capítulo 1:

Introducción

1.1 *Planteamiento del problema*

Hoy en día ante la globalización de la economía mundial, las empresas tienen que ser más competitivas para sobrevivir en el mercado actual. Para lograr que esta rivalidad sea de alguna manera equitativa, se crean estándares de calidad que deben cumplir dichas empresas para que sus productos obtengan la mejor calidad posible y así puedan ser comercializados en cualquier parte del Mundo. Una vez que el primer producto fabricado cumple con los estándares de calidad, se hace necesario controlar y repetir de manera precisa todos los materiales y procesos que dieron origen al mismo, y es por tal motivo que se busca que intervenga lo menos posible la mano del hombre, ya que la intervención del mismo se ve afectada por muchos factores externos a la producción como los son estados de ánimo, enfermedades y sentimientos, por nombrar algunos.

Las empresas para evitar que factores humanos intervengan directamente en la producción, automatizan todos o casi todos los procesos de fabricación, dejando básicamente las tareas de supervisión y mantenimiento para que sean realizadas por personas. Al automatizar los procesos de fabricación uno de los principales protagonistas son los robots, específicamente los manipuladores industriales, que por su versatilidad pueden ser adaptados a un gran número de tareas y permiten realizar las mismas con mayor rapidez, precisión y menor costo. Adicionalmente, son capaces de realizar tareas que representan un alto riesgo para la salud humana.

En Venezuela, el sector industrial no escapa al movimiento mundial de la automatización de las empresas; con la desventaja de que no es un país productor de la tecnología necesaria para tal fin. Por tal motivo, se crea una dependencia tecnológica con otros países que si poseen de la tecnología para la automatización. Adicionalmente los manipuladores robóticos son bastante costosos y por lo tanto de difícil acceso para centros educativos.

Para que estos manipuladores realmente puedan ser los protagonistas, y puedan ser adaptados a un gran número de tareas, es necesario contar con recurso humano especializado que conozca su funcionamiento y programación, y que además esté en constante actualización. Contar con trabajadores especializados en el área de manipuladores robóticos, específicamente que conozcan su funcionamiento y programación no es fácil. Para lograr que las empresas tengan personal especializado, deben entrenarlos ellos mismos, o sub-contratan de otras empresas (que también han sido entrenados) para estas labores. El problema radica en que a pesar que se puede conseguir gente capacitada, siguen siendo escasas; es por ello que hoy en día con el auge de la automatización de las empresas, y específicamente las empresas venezolanas; las Universidades deben egresar profesionales capacitados en esta área.

Debido a la dependencia tecnológica que presentan las industrias de nuestro país, y que esta tecnología provenga de otros países, nos lleva a reflexionar y a buscar que sea tomado en cuenta el estudio y desarrollo de dicha tecnología en Venezuela. A pesar de que se cuenta con personas calificadas en el área de robótica, capaces de entrenar personal; no se cuenta con los recursos materiales mínimos necesarios para impartir una enseñanza integral en ésta área. Es por estos motivos que el presente trabajo pretende comenzar a dar solución a esta problemática, a través del diseño e implementación de un manipulador robótico orientado a la enseñanza de la robótica industrial.

1.1.1 Objetivo General

Diseñar, construir y poner en funcionamiento un prototipo de un manipulador robótico de tres grados de libertad con fines educativos.

1.1.2 Objetivos específicos

Aprender y afianzar los conocimientos concernientes a la robótica industrial, específicamente manipuladores robóticos.

Aplicar y vincular los conocimientos obtenidos en el estudio de postgrado y pre-grado a la robótica industrial en las áreas de instrumentación, control de procesos, robótica de manipuladores industriales, modelado de sistemas.

Conocer y manejar a fondo la cinemática y dinámica de los manipuladores robóticos, y utilizar dichos conocimientos para controlar los movimientos y disposiciones del manipulador robótico.

Utilizar herramientas CAD (Diseño asistido por computadora) para el diseño geométrico y desarrollo del manipulador robótico.

Usar herramientas computacionales que permitan realizar simulaciones previas del comportamiento del manipulador robótico antes de que sea construido, de forma de comparar su comportamiento con el prototipo ya fabricado.

1.2 Justificación del proyecto

1.2.1 Aportes de la tesis

El diseño y puesta en funcionamiento del manipulador robótico de tres grados de libertad con fines didácticos aporta, principalmente a la Universidad de Los Andes, recursos materiales para la enseñanza integral en el área de robótica industrial. Adicionalmente con este trabajo, se continúa con la línea de investigación en el área de robots manipuladores, creando oportunidades para realizar otras investigaciones, que se convierten en oportunidades de trabajos de tesis para estudiantes de pre-grado y post-grado.

El diseño del manipulador robótico es realizado en el país, con lo que se comienza a dejar de lado la dependencia tecnológica de países extranjeros en esta área. La fabricación se realiza con materiales encontrados en su mayoría en el país, se utiliza mano de obra local para la manufactura de algunos componentes y partes. Finalmente se presenta un manipulador robótico de costo mucho menor que los que se encuentran comercialmente y que no son producidos en el país.

1.2.2 Alcance y limitaciones.

El diseño mecánico del manipulador robótico propuesto es del tipo geométrico, es decir, se realiza solamente el dimensionamiento de las partes del robot manipulador. El cálculo estructural del manipulador robótico no se realiza, ya que se asume que por las dimensiones, el bajo peso del prototipo, las bajas velocidades de funcionamiento y el uso de tipo didáctico, no se requiere de este tipo de cálculo. Sin embargo se propone el cálculo estructural para futuras investigaciones.

La fabricación del manipulador será un prototipo funcional, entendiéndose por prototipo como el primer equipo realizado en laboratorio, donde muchas de las partes fabricadas pueden no representar en su totalidad el diseño planteado, esto se debe a modificaciones que se van realizando a lo largo del proceso de fabricación y ensamblaje.

1.2.3 Contenido de los capítulos siguientes

En el segundo capítulo se tratará el estado del arte, se hablará sobre la historia de la robótica, y específicamente sobre la robótica industrial. Se introducirán los conceptos básicos necesarios para el entendimiento del tema. Finalmente se discuten los trabajos que están realizando las compañías en el área de robótica industrial, así como también el trabajo que están realizando algunas universidades alrededor del mundo; es decir, se dará a conocer las tendencias mundiales en el área de robótica industrial.

En el tercer capítulo se discutirán conceptos más profundos sobre la robótica industrial, las bases teóricas de la robótica industrial, las convenciones y herramientas matemáticas utilizadas para el

cálculo de la cinemática y dinámica de manipuladores robóticos. Se presentan las herramientas matemáticas y computacionales a utilizar para el diseño del manipulador robótico.

El cuarto capítulo muestra la propuesta realizada del manipulador robótico. Se presenta la cinemática directa e inversa y la dinámica del mismo. Se describen los sensores, actuadores y transmisores de potencia seleccionados y utilizados en el prototipo. Adicionalmente se comentan los medios de comunicación entre el computador y el prototipo del manipulador robótico. Se muestran las pruebas y simulaciones realizadas al manipulador, junto a las comparaciones con herramientas computacionales dedicadas al diseño e investigación de manipuladores robóticos.

En el quinto capítulo se presentan las conclusiones del trabajo y cuál fue el alcance real del proyecto. Se formulan recomendaciones para las personas que quieran seguir trabajando sobre este proyecto, y se presentan ideas para futuros proyectos vinculados con la línea de investigación.

WWW.BDIGITAL.ULA.VE

Capítulo 2:

Estado del arte

2.1 Definición de robots y robots manipuladores

La robótica industrial es un área de gran importancia ya que ayuda a mejorar los procesos y técnicas de fabricación, aumenta la productividad y calidad de los productos, entre otros. Por ser tan importante, es objeto de un sin número de investigaciones, y cada día se quiere mejorar en esta área para seguir aumentando la productividad de las empresas. Antes de seguir adelante con el tema, es de importancia conocer ciertos términos básicos sobre la robótica industrial.

Se parte de la definición de Robot, palabra que fue utilizada por primera vez en 1923, en una novela de ciencia ficción titulada “Rossum’s Universal Robots” y cuyo autor fue Karel Capek; y cuyo significado es el de trabajo realizado de manera forzada. Luego, con el paso del tiempo y el avance de la tecnología, el término se fue implementando y cambiando de significado según la necesidad de la época. Actualmente el significado de la palabra Robot según el diccionario de la Real Academia Española es “máquina o ingenio electrónico programable, capaz de manipular objetos y realizar operaciones antes reservadas solo a las personas”. (Real Academia Española, 2007)

Sin embargo, no existe una definición estándar y unificada para la palabra robot, y cada una de las definiciones distintas va a depender del área de la robótica en la que se esté trabajando. En el caso específico del presente trabajo, robótica de manipuladores y/o robótica industrial, existen también varios términos o definiciones, y estas van a depender de las organizaciones que las

Licencia Creative Commons:

Atribución - No Comercial - Compartir Igual 3.0 Venezuela
(CC BY-NC-SA 3.0 VE)

emitan. Está el caso del Instituto Robot de América, RIA por sus siglas en inglés (Robot Institute of America) que dice que: “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”. (Barrientos, Peñín, Balaguer, & Aracil, Fundamentos de Robótica, 2007).

Basándose en la definición de la RIA, la Organización Internacional de Estándares ISO, presenta un significado para el término de Robot manipulador industrial, el cual define como “Manipulador de 3 o más ejes, con control automático, reprogramable, multiplicación, móvil o no, destinado a ser utilizado en aplicaciones de automatización industrial. Incluye al manipulador (sistema mecánico y accionadores) y al sistema de control (software y hardware de control de potencia).” (Barrientos, Peñín, Balaguer, & Aracil, Fundamentos de Robótica, 2007).

Una vez definido el concepto de robot manipulador industrial, se pueden clasificar de varias maneras, según la generación, el área de aplicación, el tipo de actuadores, el número de ejes, por su configuración o el tipo de control. En la Tabla 1 se observa la clasificación hecha por la Federación Internacional de la Robótica IFR de los robots industriales manipuladores según el tipo de aplicación. Según la clasificación mostrada, el robot manipulador del presente trabajo estaría ubicado bajo el código 240, en el área de formación, enseñanza e investigación (Barrientos, Peñín, Balaguer, & Aracil, Fundamentos de Robótica, 2007).

2.1.1 Clasificación de los robots manipuladores

Tabla 1 Clasificación de las aplicaciones de los robots industriales manipuladores según IFR.

Código	Aplicación
000	Sin Especificar.
110	Manipulación en fundición.
130	Manipulación en moldeo de plásticos.
140	Manipulación en tratamientos térmicos.
150	Manipulación en la forja y estampación.
160	Soldadura.

Licencia Creative Commons:

Atribución - No Comercial - Compartir Igual 3.0 Venezuela
(CC BY-NC-SA 3.0 VE)

170	Aplicación de materiales.
180	Mecanización.
190	Otros procesos.
200	Montajes.
210	Paletización y empaquetado.
220	Medición, inspección, control de calidad.
230	Manipulación de materiales.
240	Formación, enseñanza e investigación.
900	Otros.

Fuente: (Barrientos, Peñín, Balaguer, & Aracil, Fundamentos de Robótica, 2007, pág. 440)

A continuación se presenta la descripción de algunas aplicaciones de los manipuladores industriales mostradas en la tabla anterior.

Empaquetado y paletizado: Estas acciones consisten en empaquetar y/o apilar los productos terminados de manera de organizarlos, para facilitar el transporte y posterior comercialización. Ver Figura 1. Los robots manipuladores comúnmente utilizados para este tipo de tareas son robots articulares de cuatro grados de libertad.

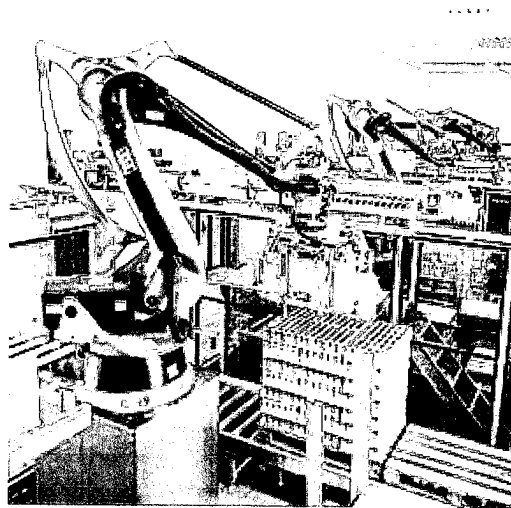


Figura 1. Manipulador para el paletizado.
Fuente: Kuka Robotics

Montaje: Entre las aplicaciones que se pueden encontrar para este grupo se tienen el ensamblado, que consiste en unir varias piezas para formar un más compleja. Los de manipulación de materiales para montaje, que se encargan de transportar material normal así como material tóxico, radioactivo o dañino para el organismo humano. Un ejemplo de un robot que manipula material se puede observar en la Figura 2.

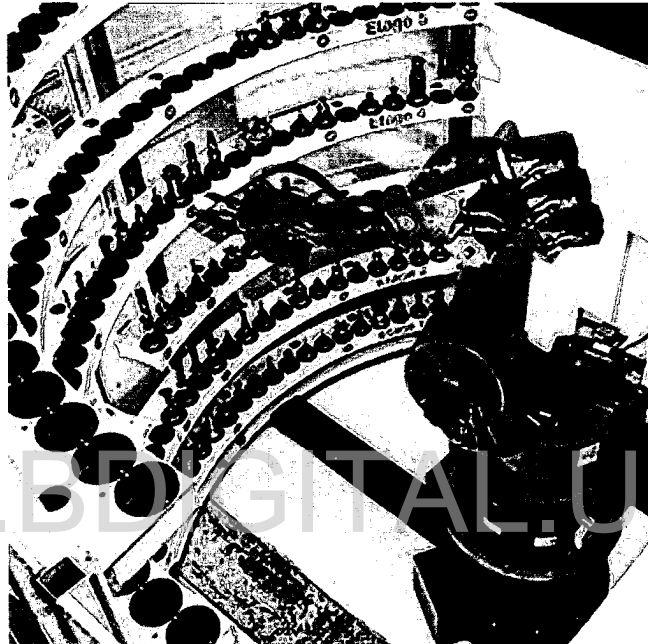


Figura 2 Robot manipulador de materiales.
Fuente: Kuka Robotics

Soldadura: En este proceso se unen dos o más partes metálicas por medio de la soldadura. Dentro de la robótica industrial, este es uno de los campos donde más se utilizan los robots manipuladores, ya que por su versatilidad se pueden utilizar para casi cualquier tipo de soldadura, puede ser por arco, por puntos, a gas, láser, etc. Siendo la industria automovilística la mayor consumidora de este tipo de robots. Por lo general estos manipuladores son de 5 ó 6 grados de libertad, para poder posicionar y ubicar el extremo soldador. Ver Figura 3.



Figura 3 Manipulador para soldadura.
Fuente: FANUC Robotics

Aplicación de materiales: Bajo este apartado se encuentran principalmente los robots manipuladores que se encargan de dar el acabado final a las superficies de trabajo, recubriéndolas con un cierto material, como el caso de pinturas, esmaltes, partículas anticorrosivas, etc. Esto con la finalidad de proteger la superficie o simplemente con fines decorativos. Al igual que los robots manipuladores para soldadura, estos robots manipuladores son utilizados principalmente por la industria automotriz, para dar el acabado final a los carros. También se encuentran los robots manipuladores que aplican materiales como sellantes y adhesivos para unir piezas, como el caso de unir el parabrisas con la carrocería. Ver Figura 4.

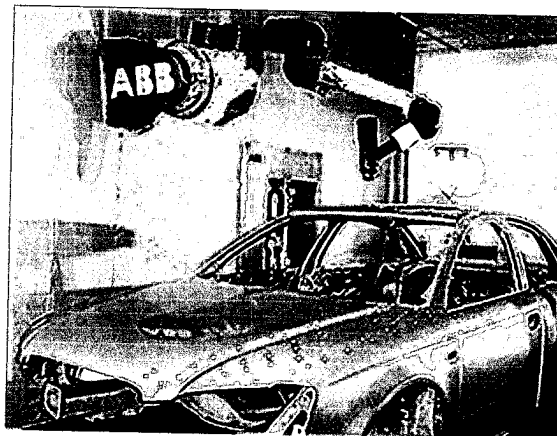


Figura 4 Manipulador para pintura de automóviles.
Fuente: ABB Robotics

Mecanización: Estos robots se encargan de fabricar piezas, mediante la remoción de material. Los procesos más comunes son el de pulido, para dar un acabado final o para la aplicación de material como pintura. El desbarbado de irregularidades en piezas plásticas o metálicas después de su fabricación. También están los que remueven más cantidad de material, como el caso de brazos manipuladores utilizados para la creación de prototipos rápidos, fabricación de moldes, etc. Ver Figura 5.



Figura 5 Manipulador para prototipado rápido.
Fuente: FANUC Robotics

Otros procesos: Los más comunes de otros procesos son el uso de robots manipuladores industriales para corte por plasma, láser o chorro de agua. Estos manipuladores son programados para cortar gran cantidad de material llevando la boquilla en el extremo, por donde se proyecta el material de corte. Pueden cortar varias capas de material a la vez, por lo que se reduce el tiempo de mecanizado. Ver Figura 6.

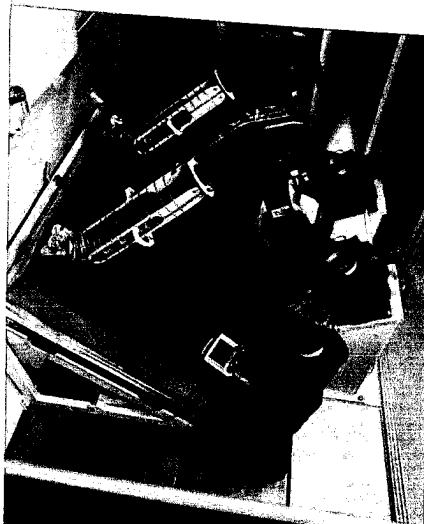


Figura 6 Manipuladores de corte por chorro de agua.
Fuente: Motoman Robotics

Finalmente se pueden encontrar otros usos que no sean específicamente en el uso de la industria de manufactura o ensamblado. Se pueden encontrar usos de robots manipuladores industriales en el área del entretenimiento, como el caso de Robocoaster, un manipulador al que se le adapta unas sillas en su extremo, y es utilizado como un “simulador de montaña rusa” donde cada persona puede programar su viaje. Ver Figura 7.

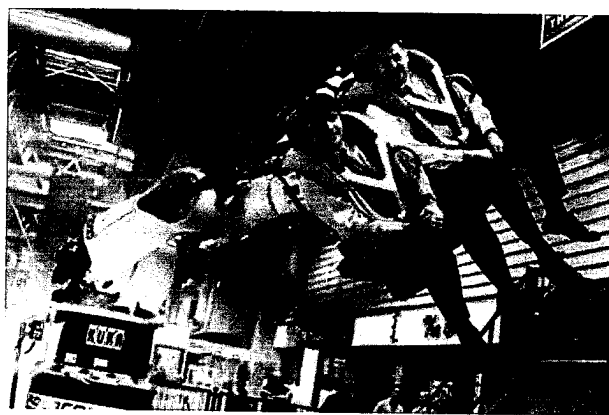


Figura 7 Robocoaster, manipulador utilizado para el entretenimiento.
Fuente: KUKA Robotics

2.1.2 Clasificación de los manipuladores según la configuración cinemática.

Robot cartesiano: Es aquel que está formado por tres ejes ortonormales entre sí. Y cuyas articulaciones son del tipo prismáticas. Presentan buena precisión y velocidad de trabajo. Ver Figura 8.

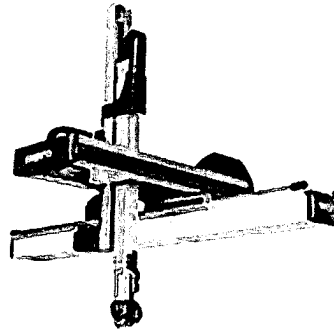


Figura 8 Robot cartesiano.

WWW.BDIGITAL.ULA.VE

Robot cilíndrico: Es similar al robot cartesiano, con la diferencia que en la base presenta la articulación es del tipo rotacional y no prismática, lo que hace que el espacio de trabajo sea un cilindro. Ver Figura 9.

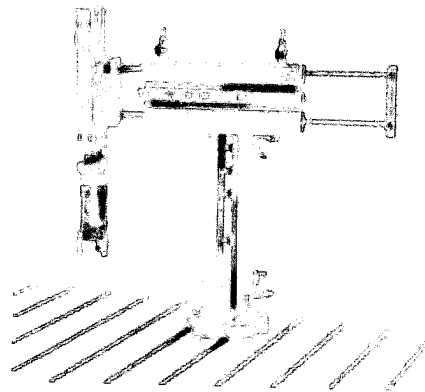


Figura 9 Robor Cilíndrico.

Licencia Creative Commons:
Atribución - No Comercial - Compartir Igual 3.0 Venezuela
(CC BY-NC-SA 3.0 VE)

Robot esférico o polar: Es una variante del robot cilíndrico, en el cual la articulación del medio es sustituida por una articulación rotacional, haciendo que el espacio de trabajo luzca como una esfera. Ver Figura 10.

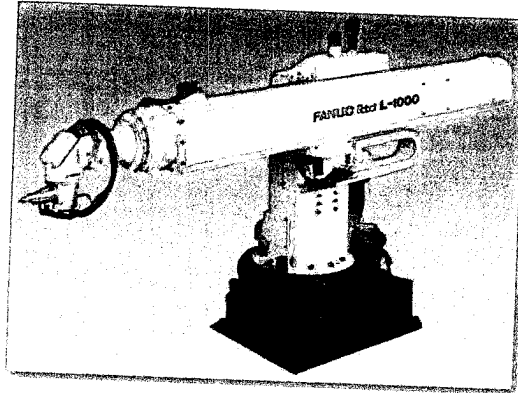


Figura 10 Robot Esférico.
Fuente: FANUC Robotics

Robot SCARA: Este robot tiene la misma configuración de articulaciones que un robot esférico, pero en una disposición diferente, es decir, en otro orden. Son de alta rapidez y precisión, y se utilizan mucho en tareas de coger y colocar. Ver Figura 11.

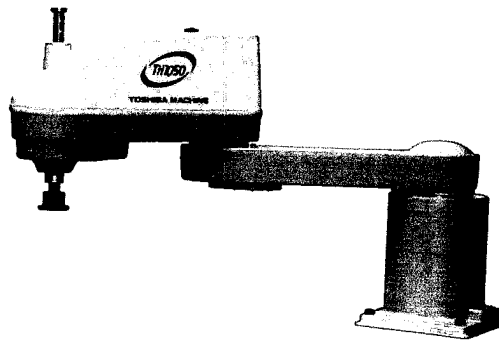


Figura 11 Robot Scara.
Fuente: Toshiba Machine

Robot articulado o antropomórfico: Este robot presenta todas sus articulaciones de tipo rotacional. Puede realizar recorridos complejos, es altamente maniobrable. Ver Figura 12

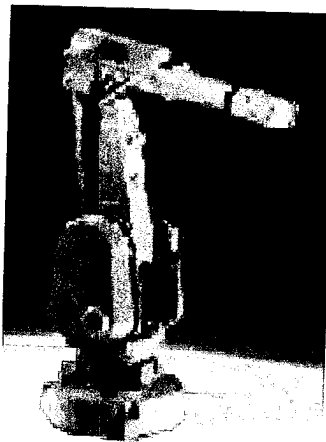


Figura 12 Robot Articulado.
Fuente: ABB Robotics

2.2 Investigación en robótica de manipuladores Industriales.

La investigación en el área de robots manipuladores nace con el inicio mismo de la robótica, cuando George C. Devol junto con Joseph F. Engelberger crean el primer manipulador llamado Unimate, en el año de 1961. Luego, en 1972 se crea en Japón la primera asociación de robótica industrial JIRA, con lo que aumenta el crecimiento de la robótica a nivel mundial. Para finales de los años 60 y comienzo de los años 70 las universidades entran en el área de investigación de la robótica industrial, de donde salen robots bastante conocidos como el brazo Stanford, ver Figura 13 que luego da origen al robot PUMA, ver Figura 14, uno de los robots más famosos en la industria y la investigación. (Barrientos, Peñín, Balaguer, & Aracil, Fundamentos de Robótica, 2007).

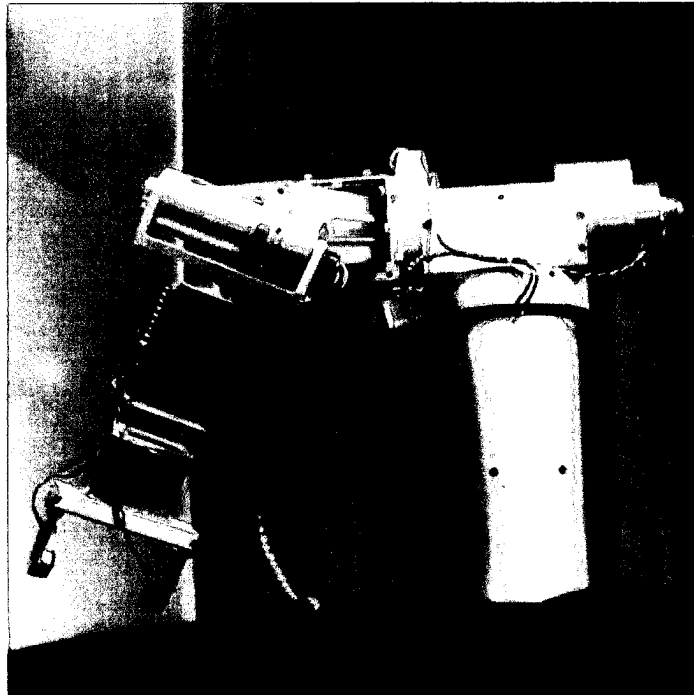


Figura 13 Manipulador Stanford.
Fuente: Universidad de Stanford

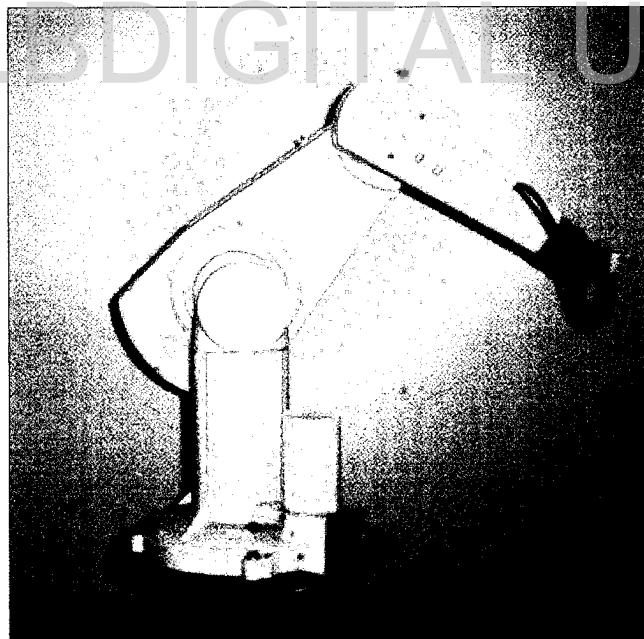


Figura 14 Manipulador robótico PUMA.
Fuente: Universidad de California

Licencia Creative Commons:
Atribución - No Comercial - Compartir Igual 3.0 Venezuela
(CC BY-NC-SA 3.0 VE)

En la actualidad son muchas las universidades y centros de investigación a nivel mundial que tienen líneas de investigación en robótica y en específico de manipuladores industriales. En el presente trabajo se hará mención de algunas de las universidades o centros de investigación con líneas de investigación en el área de robótica de manipuladores industriales.

La organización CSRIO (Commonwealth Scientific and Industrial Research Organisation) ubicada en Australia, tiene una línea de investigación en robótica de manipuladores industriales, enfocándose principalmente en el desarrollo de un software para el diseño y simulación de manipuladores robóticos. El proyecto dirigido por Peter Corke presenta un software que posee un conjunto de herramientas para el cálculo de la cinemática directa e inversa, dinámica de manipuladores, cálculo de matrices Jacobianas, y utiliza los parámetros de Denavit – Hartenberg y los parámetros Denavit-Hartenberg modificados. El software antes mencionado trabaja bajo el programa computacional de matemática llamado MATLAB ®. La línea de investigación también tienen otro software que trabaja bajo otro programa de computación llamado MAPLE ®, el cuál realiza los cálculos para el diseño de manipuladores robóticos de manera simbólica.

La Universidad Politécnica de Madrid tiene un grupo de investigación llamado Robótica y cibernética, cuenta con un gran número de colaboradores altamente reconocidos en la comunidad de habla hispana. Actualmente no trabajan directamente en la investigación de manipuladores robóticos, pero si en el área de robótica industrial con proyectos como PISA, que es un sistema de fabricación flexible a través de la división del espacio de trabajo y del tiempo en la cooperación Humano-Máquina. Algunos de los colaboradores de este grupo de investigación son autores de libros sobre robótica en idioma español, que son de consulta obligatoria cuando se quiere incursionar en el área de robótica. Actualmente están desarrollando proyectos de investigación basados en tele-operación.

Aunque sólo se mencionan dos universidades es necesario destacar que casi todas las universidades prestigiosas a nivel mundial tienen líneas de investigación en el área de robótica, Universidad de California, Stanford University, Oxofrd University, Manchester University por nombrar algunas. Además de las universidades también grandes compañías tiene laboratorios

dedicados a la investigación de robótica, como son SONY, Hitachi, Mitsubichi, Suzuki, Honda, entre otras.

La Universidad de Los Andes también cuenta con experiencia en el área de robots manipuladores industriales, aunque con trabajos aislados y no con un grupo de investigación dedicado a tal fin. Los trabajos realizados abarcan varias áreas como el diseño, control, simulación, modelado de sistemas, entre otros. Entre los trabajos realizados se pueden mencionar

El trabajo “Desarrollo de una librería de modelos de brazos manipuladores y evaluación de leyes de control” elaborado por Rhyna Herrera y Pablo Lischinsky en el año de 1998 desarrollan una librería con tres modelos de robots, uno tipo Planar, uno tipo SCARA y por último un Cincinnati T3. Se evalúan cinco tipos de algoritmo sobre el modelo Cincinnati T3. Los dos primeros algoritmos son de tipo descentralizado, un PD y un PD con pre-compensación. El tercer algoritmo utilizado se conoce como dinámica inversa, y los dos últimos algoritmos de control son no lineales basados en pasividad. Para probar los algoritmos de control, se crean tres trayectorias que debe seguir el manipulador, y se simulan con y sin carga. (Herrera & Lischinsky, 1998).

“Diseño y construcción de un robot scara controlado por PLC”. Carlos Cadenas. Y Pedro O. Mora en el año de 2001 presenta el diseño de un robot de tres grados de libertad tipo SCARA, donde se utilizan motores de paso eléctrico para cada una de las articulaciones, y controlados independientemente mediante el uso de un PLC.

Existen otros trabajos previos como “Diseño y construcción de un elemento terminal para un robot” por Efrén Blaque en el año de 1987. “Estudio teórico de un robot de dos grados de libertad con flexibilidad”, de Alvaro Francisco Cañizales Ponte y Jean Claude Dulhoste en el año de 1993 y “Diseño de un control adaptativo para un robot manipulador con tres grados de libertad” de Jorge Luis Valera Briceño en el año de 1992.

Finalmente con el presente trabajo, en el año de 2006 se crea una línea de investigación ante el Consejo de Desarrollo Científico Humanístico y Tecnológico (CDCHT), dedicada al área de

robótica y dirigida por el Dr. Iñaki Aguirre, profesor de la Escuela de Ingeniería de Sistemas de la Universidad de Los Andes. Paralelamente al presente proyecto se trabajó con el desarrollo de una herramienta computacional para la simulación y programación del manipulador robótico aquí propuesto. Actualmente se está trabajando en un proyecto para realizar la comunicación entre la herramienta computacional desarrollada y el prototipo físico del manipulador.

WWW.BDIGITAL.ULA.VE

Licencia Creative Commons:
Atribución - No Comercial - Compartir Igual 3.0 Venezuela
(CC BY-NC-SA 3.0 VE)

Capítulo 3:

Generalidades de los manipuladores

Existen múltiples herramientas matemáticas y computacionales para el diseño de manipuladores robóticos, unas de uso más general, que son las que se utilizan con más frecuencia; y otras herramientas que, según sea el caso, son de uso más específico y por lo tanto utilizadas con menos frecuencia. En este capítulo se presentan las herramientas matemáticas de uso general en el diseño de manipuladores robóticos, específicamente las herramientas utilizadas para el diseño del manipulador robótico del presente trabajo. Posteriormente se hace referencia a las herramientas computacionales utilizadas para el cálculo matemático y herramientas computacionales para el diseño mecánico.

3.1 Herramientas matemáticas necesarias en el uso de manipuladores

Un manipulador robótico puede cumplir diversas tareas, según haya sido programado y según la aplicación. El cumplimiento de estas tareas implica el movimiento del manipulador de una posición a otra, y adicionalmente la orientación que debe tomar según la labor a realizar. Para que pueda moverse de un lugar a otro debe estar referenciado espacialmente, es decir, conocer en todo momento la posición en que se encuentra el manipulador; para ello, y por lo general, se toma la base del manipulador como punto de referencia para la ubicación. Para la orientación, de igual manera, se toma el mismo punto de referencia que para la ubicación, la base.

Con la base como punto de referencia para la ubicación y orientación del manipulador, se hace necesario contar con herramientas matemáticas que permitan predecir las nuevas posiciones y orientaciones en el espacio de trabajo, tanto del manipulador como de los objetos que interactúan con él. Estas herramientas (métodos numéricos, ecuaciones) deben permitir obtener relaciones espaciales sencillas para que al ser programadas no consuman muchos recursos computacionales.

Seguidamente se dan a conocer las herramientas matemáticas utilizadas en el diseño de manipuladores robóticos, comenzando con la ubicación espacial, para luego pasar a las transformadas homogéneas que permiten calcular ubicación y orientación espacial al mismo tiempo. Las herramientas que se presentan a continuación no son sólo de uso exclusivo en la robótica, sino que pueden ser utilizadas en problemas de ubicación y orientación espacial en general, incluso son muy utilizadas en el campo de los gráficos generados por computador (Barrientos, Peñín, Balaguer, & Aracil, Fundamentos de Robótica, 2007).

3.1.1 Posicionamiento

El posicionamiento permite colocar el extremo del manipulador en una ubicación espacial específica, es decir, sobre unas coordenadas específicas y teniendo por lo general como referencia la base del manipulador. Una vez posicionado el extremo del manipulador se deberá orientar para que realice su tarea.

3.1.1.1 Sistemas de referencia cartesianos.

Partiendo de que se tiene el punto de referencia, es necesario definir un sistema de referencia asociado al cuerpo rígido. Uno de los sistemas de referencia más utilizado es el cartesiano, que consiste o se define por medio de tres planos ortogonales, y se representa por un sistema dextrógiro de ejes, que en el caso de ser una representación bidimensional (dos dimensiones) se utilizan dos ejes, y en el caso tridimensional (tres dimensiones) se utilizan tres ejes; ambos casos se muestran en la Figura 15 y Figura 16.

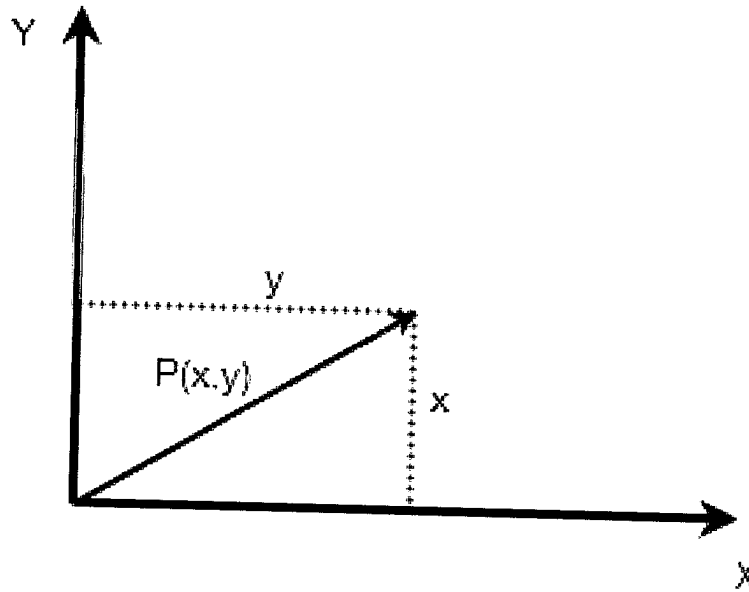


Figura 15 Sistema cartesiano 2D

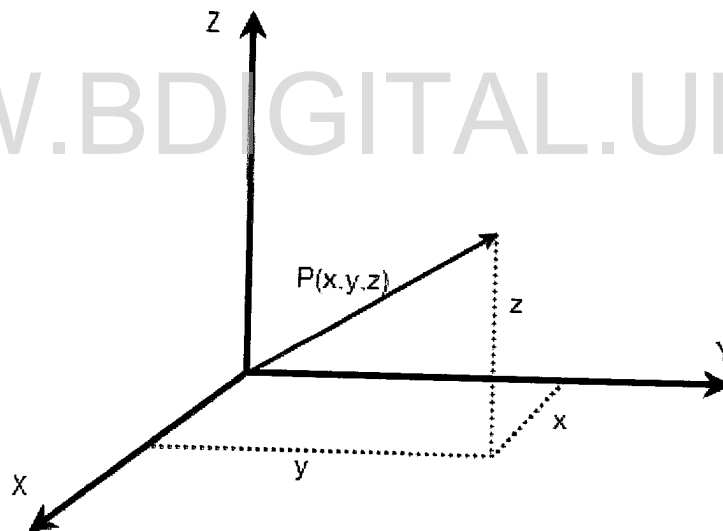


Figura 16 Sistema cartesiano 3D

El sistema de referencia cartesiano, permite ubicarse en el espacio, pero no sirve de nada por sí solo, es necesario contar con un sistema de coordenadas que permitan cuantificar la posición y orientación; para ello existen tres sistemas de coordenadas que se utilizan con el sistema de

Licencia Creative Commons:
Atribución - No Comercial - Compartir Igual 3.0 Venezuela
(CC BY-NC-SA 3.0 VE)

referencia cartesiano: coordenadas cartesianas, coordenadas polares y cilíndricas, y las coordenadas esféricas.

Las coordenadas cartesianas hacen referencia a un punto de trabajo ubicado en el espacio, con el que se forma un vector junto con el punto de origen del sistema. Este vector es proyectado sobre los planos de referencia y sobre los ejes de referencia, con lo que se obtienen las componentes de las coordenadas que forman el vector posición. Para el caso de la Figura 15 y Figura 16, el vector P está formado por el origen y por el punto a , cuyas coordenadas cartesianas son $a(x,y)$ para el caso bidimensional y $a(x,y,z)$ para el caso tridimensional.

Sobre el mismo sistema de referencia cartesiano se desarrollan las coordenadas polares y cilíndricas. El vector P viene definido para el caso bidimensional por la distancia desde el origen hasta el punto $a(r)$, y el ángulo que forma con respecto al eje $X(\theta)$, y cuya representación está dada por $P(r,\theta)$ ver Figura 17; para el caso tridimensional viene definido de igual manera por la distancia del origen al punto $a(r)$, por el ángulo que forma con respecto al eje $X(\theta)$ y la proyección sobre el eje $Z(z)$, y cuya representación es $P(r,\theta,z)$ ver Figura 18.

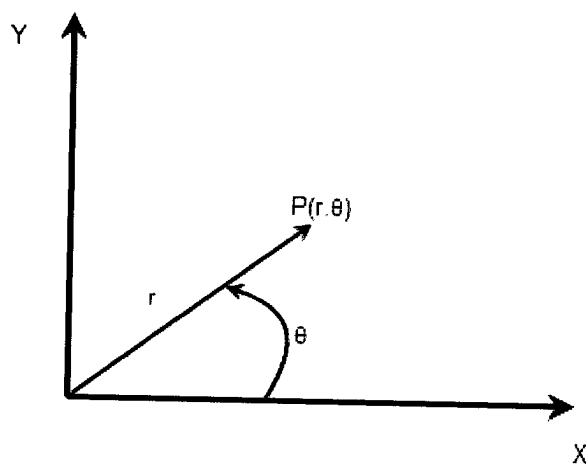


Figura 17 Coordenadas polares

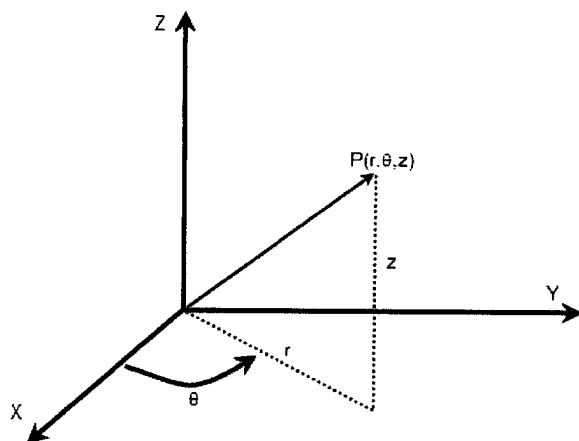


Figura 18 Coordenadas cilíndricas

Por último, y utilizando el sistema de referencia cartesiano se tienen las coordenadas esféricas que son utilizadas únicamente para ubicación tridimensional. En este caso, el vector P está definido por la distancia desde el origen al punto $a(r)$, el ángulo que forma la proyección en el plano XY con el eje $X(\theta)$ y el ángulo que forma con el eje $Z(\phi)$, y cuya representación final es $P(r, \theta, \phi)$, ver Figura 19.

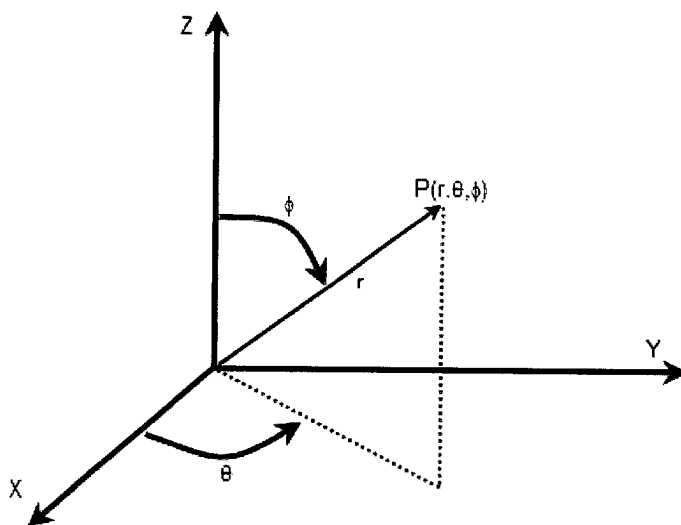


Figura 19 Coordenadas esféricas

3.1.2 Orientación

La orientación es la disposición espacial del elemento terminal del manipulador con respecto a un sistema de referencia, para realizar una tarea; geoméricamente son los ángulos que forma el eje de coordenadas del elemento terminal respecto a los ejes coordenados del origen.

Para definir totalmente la ubicación en el espacio de un manipulador industrial se hace necesario conocer adicionalmente la orientación. Esto permite no sólo ubicar un elemento terminal o herramienta del robot, sino que además hace posible definir como se orienta para que realice su función o tarea.

Definir la orientación de un cuerpo en el espacio tridimensional requiere del uso de tres grados de libertad o tres componentes linealmente independientes. Una manera sencilla de describir la orientación es asignar un nuevo sistema de coordenadas al cuerpo, ubicado coincidentemente con el origen del sistema de referencia principal, y luego estudiar la relación espacial existente entre los dos sistemas (Barrientos, Peñín, Balaguer, & Aracil, Fundamentos de Robótica, 1997).

Existen varios métodos para conocer la orientación de un cuerpo en el espacio, entre los que se encuentran: Ángulos de Euler, par de rotación, cuaternios y las matrices de rotación. Este último método es el más utilizado dentro de la robótica industrial debido a la simplicidad que ofrece el uso del álgebra matricial.

El método de las matrices de rotación consiste en conseguir una matriz que define la orientación entre el sistema de referencia principal y el sistema de referencia nuevo creado para el estudio. Esta matriz, llamada matriz de rotación, permite además transformar las coordenadas de un sistema de referencia al otro, y es el más utilizado en el cálculo de manipuladores robóticos (Barrientos, Peñín, Balaguer, & Aracil, Fundamentos de Robótica, 1997).

Para comprender como funciona el método de las matrices de rotación se asume que los dos sistemas de referencia se encuentran en el mismo origen y con la misma orientación, y que a

partir de ahora serán identificados como OXYZ para el sistema de coordenadas principal y cuyo vector unitario será (i_x, j_y, k_z) ; el sistema de coordenadas nuevo será identificado como OUVW con vector unitario (i_u, j_v, k_w) , como se muestra en la Figura 20.

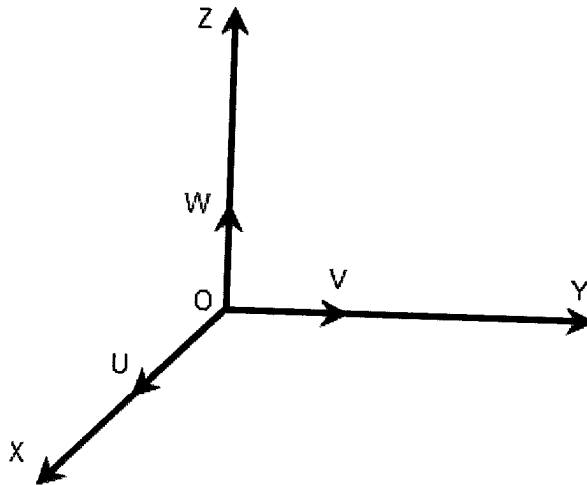


Figura 20 Sistemas de coordenadas principal OXYZ y móvil OUVW

Representar un vector en los sistemas de coordenadas principal y móvil será de la forma:

$$\begin{aligned} P_{xyz} &= [p_x, p_y, p_z]^T = p_x \cdot i_x + p_y \cdot j_y + p_z \cdot k_z \\ P_{uvw} &= [p_u, p_v, p_w]^T = p_u \cdot i_u + p_v \cdot j_v + p_w \cdot k_w \end{aligned} \quad (1)$$

Y cuya representación se puede simplificar como:

$$\begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix} = R \begin{bmatrix} p_u \\ p_v \\ p_w \end{bmatrix} \quad (2)$$

Donde R es la matriz de rotación que define la orientación de OUVW respecto de OXYZ, y es una matriz ortonormal, tal que la inversa es igual a la transpuesta: $R^{-1}=R^T$ (Torres, Pomares, Gil, Puente, & Aracil, 2002).

$$R = \begin{bmatrix} i_x \cdot i_u & i_x \cdot j_v & i_x \cdot k_w \\ j_y \cdot i_u & j_y \cdot j_v & j_y \cdot k_w \\ k_z \cdot i_u & k_z \cdot j_v & k_z \cdot k_w \end{bmatrix} \quad (3)$$

La matriz de rotación R sólo muestra la orientación de los sistemas girados sobre uno de los ejes principales. Por lo tanto sólo se tendrán las representaciones sobre los ejes X, Y y Z; cuyos ángulos de giro serán α, ϕ, θ respectivamente, ver Figura 21, Figura 22 y Figura 23.

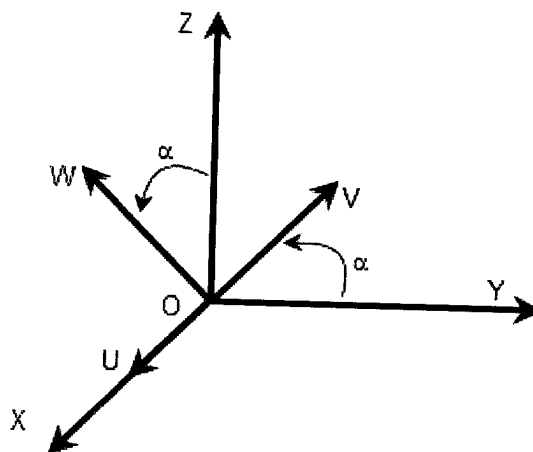


Figura 21 Sistema coordenado girado sobre el eje X

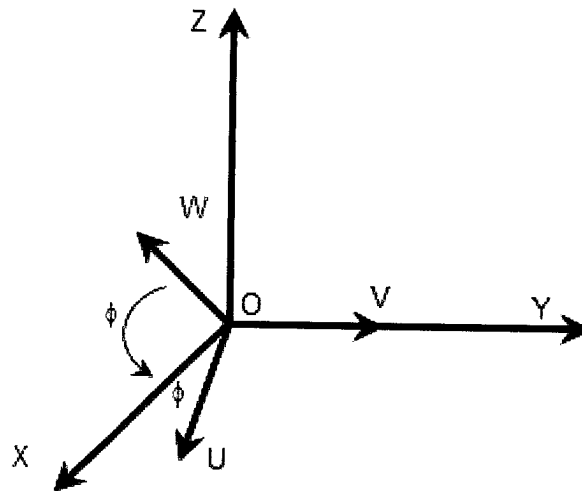


Figura 22 Sistema coordenado girado sobre eje Y

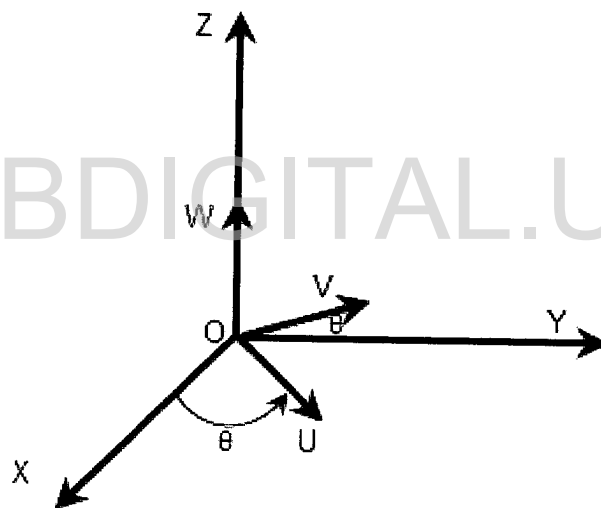


Figura 23 Sistema coordenado girado sobre eje Z

Las matrices de rotación, llamadas matrices de cosenos o matrices básicas de rotación; vienen representadas para cada uno de los ejes por:

$$\begin{aligned}
 R(x, \alpha) &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \alpha & -\sin \alpha \\ 0 & \sin \alpha & \cos \alpha \end{bmatrix} \\
 R(y, \phi) &= \begin{bmatrix} \cos \phi & 0 & \sin \phi \\ 0 & 1 & 0 \\ -\cos \phi & 0 & \cos \phi \end{bmatrix} \\
 R(z, \theta) &= \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix}
 \end{aligned} \tag{4}$$

Como se menciona anteriormente, las matrices de rotación sólo permiten un movimiento a la vez, pero se pueden construir más complejos a través de la unión de varios movimientos por separado. Por ejemplo, si se quieren tres rotaciones, una sobre el eje OX, otra sobre el eje OY y una última sobre el eje OZ, la rotación final se puede definir como:

$$\begin{aligned}
 T &= R(x, \alpha)R(y, \phi)R(z, \theta) = \\
 &\begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \alpha & -\sin \alpha \\ 0 & \sin \alpha & \cos \alpha \end{bmatrix} \begin{bmatrix} \cos \phi & 0 & \sin \phi \\ 0 & 1 & 0 \\ -\cos \phi & 0 & \cos \phi \end{bmatrix} \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} = \\
 &\begin{bmatrix} \cos \phi \cos \theta & -\cos \phi \sin \theta & \sin \phi \\ \sin \alpha \sin \phi \cos \theta + \cos \alpha \sin \theta & -\sin \alpha \sin \phi \sin \theta + \cos \alpha \cos \theta & -\sin \alpha \cos \phi \\ -\cos \alpha \sin \phi \cos \theta + \sin \alpha \sin \theta & \cos \alpha \sin \phi \sin \theta + \sin \alpha \cos \theta & \cos \alpha \cos \phi \end{bmatrix}
 \end{aligned} \tag{5}$$

3.1.3 Matrices y Coordenadas Homogéneas

En los apartados anteriores se describen formas de posicionar u orientar un objeto en el espacio, pero en la robótica industrial es necesario que esa orientación y posicionamiento se hagan de manera conjunta. En 1969 Forest introduce las coordenadas homogéneas y la matriz de transformación homogénea, con lo que se podía resolver a través de matrices los problemas de

los gráficos por computadora; y son estas mismas coordenadas y la matriz, las que se utilizan hoy en día para resolver los problemas de posición y orientación de robots manipuladores industriales (Torres, Pomares, Gil, Puente, & Aracil, 2002).

Las coordenadas homogéneas representan la localización en el espacio de un objeto a través de un espacio $(n+1)$ -dimensional, de manera que representar un vector $p(x, y, z)$ en coordenadas homogéneas será de la forma $p(wx, wz, wz, w)$, donde w representa un factor de escala y que por lo general es uno (1). (Barrientos, Peñín, Balaguer, & Aracil, Fundamentos de Robótica, 2007).

La matriz de transformación homogénea denominada T , representa la transformación de un vector de coordenadas homogéneas a otro sistema de coordenadas. Es de dimensión 4×4 y se compone de cuatro sub-matrices, una sub-matriz de rotación $R_{3 \times 3}$, de posición $P_{3 \times 1}$, de perspectiva $f_{1 \times 3}$ y una última de escalado $w_{1 \times 1}$.

$$T = \begin{bmatrix} \text{Rotación} & \text{Posición} \\ \text{Perspectiva} & \text{Escalado} \end{bmatrix} = \begin{bmatrix} R_{3 \times 3} & P_{3 \times 1} \\ f_{1 \times 3} & w_{1 \times 1} \end{bmatrix} \quad (6)$$

De las cuatro sub-matrices realmente interesan las de rotación y la de traslación debido a que en la robótica industrial se trabaja directamente con posiciones y orientaciones reales en el espacio tridimensional de objetos reales (Torres, Pomares, Gil, Puente, & Aracil, 2002). Por tal motivo, la sub-matriz de perspectiva será cero (0) en sus tres componentes; y la sub-matriz de escalado tomará el valor de uno (1), por ser escala real 1:1. Resultando la matriz de transformación homogénea T :

$$T = \begin{bmatrix} \text{Rotación} & \text{Posición} \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} R_{3 \times 3} & P_{3 \times 1} \\ 0 & 1 \end{bmatrix} \quad (7)$$

La matriz de transformación homogénea T , ver Ecuación (7), representa la nueva posición y orientación de un sistema $O'UVW$ con respecto a un sistema de referencia $OXYZ$. Con la matriz T se puede conocer las coordenadas de un vector $r(rx,ry,rz)$ del sistema $OXYZ$ a partir de las coordenadas $r(ru,rv,rw)$ del sistema $O'UVW$; y de manera viceversa.

Cabe aclarar que aún y cuando la matriz de transformación homogénea T representa la orientación y posición, lo hace a través de traslaciones y rotaciones con respecto a un sistema de referencia, por lo tanto es necesario definir el orden en que lo realizan, pues se tratan de transformaciones espaciales no conmutativas, es decir, no es lo mismo realizar primero una rotación y luego una traslación, que una traslación y luego una rotación; pues el resultado final es distinto.

Los movimientos más comunes son los de una rotación seguida de una traslación o viceversa, en ambos casos, para realizar los cálculos, se colocan en sentido inverso para realizar la operación, es decir, se pre-multiplican sobre las transformaciones ya aplicadas; para hacer una rotación sobre el eje OX y luego una traslación de vector $P(x,y,z)$ se tiene que:

$$T(P)Rotx(\phi) = \begin{bmatrix} 1 & 0 & 0 & Px \\ 0 & \cos \phi & -\sin \phi & Py \\ 0 & \sin \phi & \cos \phi & Pz \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (8)$$

Mientras que realizar una traslación del vector $P(x,y,z)$ y luego una rotación:

$$Rotx(\phi)T(P) = \begin{bmatrix} 1 & 0 & 0 & Px \\ 0 & \cos \phi & -\sin \phi & Py \cos \phi - Pz \sin \phi \\ 0 & \sin \phi & \cos \phi & Py \sin \phi + Pz \cos \phi \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (9)$$

Estos cálculos se realizan con respecto al sistema de referencia fijo. En caso de realizar los cálculos con respecto al sistema de referencia móvil, es decir, con respecto a la última localización del sistema de referencia transformado, estas se post-multiplican respecto a las aplicadas anteriormente.

La matriz homogénea de transformación T , representa las transformaciones: rotaciones y traslaciones; de vectores de un sistema de referencia móvil $O'UVW$ con respecto a uno fijo $OXYZ$. Esta matriz suele expresarse de la forma:

$$T = \begin{bmatrix} n_x & o_x & a_x & P_x \\ n_y & o_y & a_y & P_y \\ n_z & o_z & a_z & P_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (10)$$

Donde (n,o,a) es una sub-matriz ortonormal que representa la orientación y P es el vector que representa la posición.

En muchos casos, se hace necesario utilizar la matriz de transformación inversa, que se representa por:

$$T^{-1} = \begin{bmatrix} n_x & n_y & n_z & -n^T P \\ o_x & o_y & o_z & -o^T P \\ a_x & a_y & a_z & -a^T P \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (11)$$

Para representar movimientos complejos, más allá de una rotación y una traslación, se descomponen los movimientos en varias rotaciones y traslaciones consecutivas, que serán

expresadas en función del sistema de coordenado deseado. En el caso de que se quiera expresar en función del sistema de coordenado OXYZ se coloca el primer movimiento, y luego se pre-multiplica cada uno de los movimientos siguientes. Por ejemplo, realizar un giro sobre el eje $OX(\alpha)$, un giro sobre el eje $OY(\theta)$ y finalmente un giro sobre el eje $OZ(\phi)$, se tendrá que:

$$\begin{aligned}
 T &= Rotz(\phi)Roty(\theta)Rotx(\alpha) = \\
 &\begin{bmatrix} \cos \theta & -\sin \theta & 0 & 0 \\ \sin \theta & \cos \theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos \phi & 0 & \sin \phi & 0 \\ 0 & 1 & 0 & 0 \\ -\sin \phi & 0 & \cos \phi & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \alpha & -\sin \alpha & 0 \\ 0 & \sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \\
 &\begin{bmatrix} \cos \phi \cos \theta & -\sin \theta \cos \alpha - \cos \theta \sin \phi \sin \alpha & \sin \theta \sin \alpha + \cos \theta \sin \phi \cos \alpha & 0 \\ \sin \theta \cos \phi & \cos \phi \cos \alpha + \sin \theta \sin \phi \sin \alpha & -\cos \theta \sin \alpha + \sin \theta \sin \phi \cos \alpha & 0 \\ -\sin \phi & \cos \phi \sin \alpha & \cos \alpha \cos \phi & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (12)
 \end{aligned}$$

Las transformaciones también se pueden expresar en función del sistema coordenado que está en movimiento OUVW, en este caso las matrices de transformación si se colocan en el orden de los movimientos; para el mismo caso anterior, pero con respecto al sistema de referencia OUVW se tiene que:

$$\begin{aligned}
 T &= Rotx(\alpha)Roty(\theta)Rotz(\phi) = \\
 &\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \alpha & -\sin \alpha & 0 \\ 0 & \sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos \phi & 0 & \sin \phi & 0 \\ 0 & 1 & 0 & 0 \\ -\sin \phi & 0 & \cos \phi & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos \theta & -\sin \theta & 0 & 0 \\ \sin \theta & \cos \theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \\
 &\begin{bmatrix} \cos \phi \cos \theta & -\cos \phi \sin \theta & \sin \phi & 0 \\ \sin \alpha \sin \phi \cos \theta + \cos \alpha \sin \theta & -\sin \alpha \sin \phi \sin \theta + \cos \alpha \cos \theta & -\sin \alpha \cos \phi & 0 \\ -\cos \alpha \sin \phi \cos \theta + \sin \alpha \sin \theta & \cos \alpha \sin \phi \sin \theta + \sin \alpha \cos \theta & \cos \alpha \cos \phi & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (13)
 \end{aligned}$$

3.1.4 Cinemática de robots

La cinemática de robots se encarga del estudio de los movimientos de los robots, en nuestro caso en particular, en el estudio de los movimientos de manipuladores robóticos. En la cinemática de manipuladores se describe el movimiento espacial en función del tiempo, junto con las relaciones de posicionamiento y orientación del extremo del manipulador; en ningún momento se estudian las fuerzas que interactúan en dichos movimientos.

El estudio cinemático de los manipuladores se puede realizar de dos maneras, el estudio cinemático directo y el estudio cinemático inverso. El estudio cinemático directo determina la posición final y orientación del manipulador basándose en su configuración geométrica y en los valores que toman los actuadores en las articulaciones en un determinado momento. El estudio cinemático inverso determina los valores que deben tomar los actuadores en las articulaciones, para que un manipulador, con una configuración geométrica establecida, tome una posición y orientación espacial conocida.

3.1.5 Cinemática directa

Al considerar el manipulador robótico como una secuencia cinemática de eslabones rígidos, unidos por articulaciones; se puede asignar un sistema de referencia fijo, ubicado generalmente en la base del manipulador; a fin de localizar cada uno de los eslabones en el espacio.

Una vez ubicados espacialmente los eslabones del manipulador, el problema de la cinemática directa se puede resolver de varias maneras; una de las dos más utilizadas es mediante métodos geométricos, en donde la relación entre las articulaciones se deduce mediante el uso de la trigonometría; con la dificultad que a mayor número de eslabones se hacen más complejos los cálculos. La segunda manera más utilizada para la resolución del problema de la cinemática directa se reduce a encontrar la matriz de transformación homogénea T , para relacionar la posición y orientación del extremo del manipulador con el sistema coordinado de referencia ubicado en la base.

3.1.5.1 Matrices de transformación homogénea en la cinemática directa

El uso de las transformaciones homogéneas se basa en asignar a cada uno de los eslabones un sistema de referencia propio, incluyendo la base del robot y el extremo. Existe una transformación homogénea, compuesta de rotaciones y traslaciones básicas, que permiten pasar del sistema de referencia de la base al sistema de referencia del extremo; y que está dada en función de los parámetros articulares.

La matriz de transformación homogénea asociada a la posición y orientación relativa entre dos eslabones se denomina ${}^{i-1}A_i$. En el caso de que dos eslabones sean consecutivos, la matriz A representará la transformación entre el eslabón anterior y el eslabón en estudio, por ejemplo si se estudia el primer eslabón se tiene que 0A_1 , donde 0 representa la base y 1 el primer eslabón. La matriz de transformación homogénea no sólo representa los movimientos entre eslabones consecutivos, además, representa los movimientos y transformaciones existentes en una cadena cinemática consecutiva, y viene representada por 0A_k , donde k representa el último eslabón utilizado; y la matriz de transformación homogénea total es el producto resultante de las matrices ${}^{i-1}A_i$ intermedias. Cuando se consideran todos los grados de libertad de un manipulador, entonces la matriz 0A_k se denomina matriz de transformación homogénea T . Así por ejemplo, un manipulador con cuatro grados de libertad tendrá definida su matriz de transformación T por:

$$T = {}^0A_4 = {}^0A_1 {}^1A_2 {}^2A_3 {}^3A_4 \quad (14)$$

3.1.5.2 Algoritmo Denavit Hartenberg

En el desarrollo del problema de la cinemática directa, se puede asignar cualquier sistema de referencia a cada uno de los eslabones, es común utilizar en la robótica la representación de Denavit-Hartenberg (D-H) (Barrientos, Peñín, Balaguer, & Aracil, Fundamentos de Robótica, 2007).

Denavit y Hartenberg en el año de 1955, propusieron un algoritmo, donde establecen como colocar los sistemas de coordenadas $\{S_i\}$ ligados a cada eslabón, y luego sistematizan la obtención de las ecuaciones cinemáticas mediante el uso de cuatro transformaciones básicas que dependen de la geometría del eslabón. Estas transformaciones son una sucesión de rotaciones y traslaciones que permiten relacionar el sistema de referencia $i-1$ con el sistema de referencia i ; y que se mencionan a continuación:

1. Rotación alrededor del eje z_{i-1} un ángulo θ_i
2. Traslación a lo largo del eje z_{i-1} una distancia d_i ; vector $d_i(0,0,d_i)$
3. Traslación a lo largo de x_i una distancia a_i ; vector $a_i(a_i,0,0)$
4. Rotación alrededor del eje x_i un ángulo α_i

De este modo se tiene que:

$${}^{i-1}A_i = \text{Rot}_z(\theta_i)T(0,0,d)T(a_i,0,0)\text{Rot}_x(\alpha_i) \quad (15)$$

Y realizando el producto entre las matrices se tiene que:

$${}^{i-1}A_i = \begin{bmatrix} \cos\theta_i & -\sin\theta_i & 0 & 0 \\ \sin\theta_i & \cos\theta_i & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & a_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\alpha_i & -\sin\alpha_i & 0 \\ 0 & \sin\alpha_i & \cos\alpha_i & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \cos\theta_i & -\cos\alpha_i \sin\theta_i & \sin\alpha_i \sin\theta_i & a_i \cos\theta_i \\ \sin\theta_i & \cos\alpha_i \cos\theta_i & -\sin\alpha_i \cos\theta_i & a_i \sin\theta_i \\ 0 & \sin\alpha_i & \cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (16)$$

Donde $\theta_i, d_i, a_i, \alpha_i$ son los parámetros Denavit-Hartenberg (DH) del eslabón i . Y dependen únicamente de las características geométricas de los eslabones y las articulaciones que los unen.

θ_i Es el ángulo que forman los ejes x_{i-1} y x_i medido en un plano perpendicular al eje z_{i-1} , utilizando la regla de la mano derecha. Se trata de un parámetro variable en articulaciones giratorias.

d_i Es la distancia a lo largo del eje z_{i-1} desde el origen del sistema de coordenadas $(i-1)$ -ésimo hasta la intersección del eje z_{i-1} con el eje x_i . Se trata de un parámetro variable en articulaciones prismáticas.

a_i Es la distancia a lo largo del eje x_i que va desde la intersección del eje z_{i-1} con el eje x_i hasta el origen del sistema i -ésimo, en el caso de articulaciones giratorias. En el caso de articulaciones prismáticas, se calcula como la distancia más corta entre los ejes z_{i-1} y z_i .

α_i Es el ángulo de separación del eje z_{i-1} y el eje z_i , medido en un plano perpendicular al eje x_i utilizando la regla de la mano derecha. (Barrientos, Peñín, Balaguer, & Aracil, Fundamentos de Robótica, 2007) (Pág. 126)

Adicionalmente a los parámetros Denavit-Hartenberg, es necesario definir los sistemas de referencia $\{S_{i-1}\}$ y $\{S_i\}$, para utilizarlos en la matriz de transformación ${}^{i-1}A_i$, para ello se deben haber seleccionado siguiendo el algoritmo planteado también por Denavit-Hartenberg:

1. Numerar los eslabones comenzando con 1 (primer eslabón móvil de la cadena) y acabando con n (último eslabón móvil). Se numerará como eslabón 0 a la base fija del robot.
2. Numerar cada articulación comenzando por 1 (correspondiente al primer grado de libertad) y acabando en n .
3. Localizar el eje de cada articulación. Si esta es rotativa, el eje será su propio eje de giro. Si es prismática, será el eje a lo largo del cual se produce el desplazamiento.
4. Para i de 0 a $n-1$ situar el eje z_i sobre el eje de la articulación $i+1$.
5. Situar el origen del sistema de la base $\{S_0\}$ en cualquier punto del eje z_0 . Los ejes x_0 e y_0 se situarán de modo que formen un sistema dextrógiro con z_0 .
6. Para i de 1 a $n-1$, situar el origen del sistema $\{S_i\}$ (solidario al eslabón i) en la intersección del eje z_i con la línea normal común a z_{i-1} y z_i . Si ambos ejes

- se cortasen se situaría $\{S_i\}$ en el punto de corte. Si fuesen paralelos $\{S_i\}$ se situaría en la articulación $i+1$.
7. Situar x_i en la línea normal común a z_{i-1} y z_i .
 8. Situar y_i de modo que forme un sistema dextrógiro con x_i y z_i .
 9. Situar el sistema $\{S_n\}$ en el extremo del robot de modo que z_n coincida con la dirección de z_{n-1} y x_n sea normal a z_{n-1} y z_n .
 10. Obtener θ_i como el ángulo que hay que girar en torno a z_{i-1} para que x_{i-1} y x_i queden paralelos.
 11. Obtener d_i como la distancia, medida a lo largo de z_{i-1} , que habría que desplazar $\{S_{i-1}\}$ para que x_i y x_{i-1} quedasen alineados.
 12. Obtener a_i como la distancia medida a lo largo de x_i (que ahora coincidiría con x_{i-1}) que habría que desplazar el nuevo $\{S_{i-1}\}$ para que su origen coincidiese con $\{S_i\}$.
 13. Obtener α_i como el ángulo que habría que girar en torno a x_i para que el nuevo $\{S_{i-1}\}$ coincidiese totalmente con $\{S_i\}$.
 14. Obtener las matrices de transformación ${}^{i-1}A_i$ definidas en la ecuación (16).
 15. Obtener la matriz de transformación que relaciona al sistema de la base con el del extremo del robot $T = {}^0A_1 \cdot {}^1A_2 \cdot \dots \cdot {}^{n-1}A_n$.
 16. La matriz T define la orientación (submatriz de rotación) y posición (submatriz de traslación) del extremo referido a la base, en función de las n coordenadas articulares. (Barrientos, Peñín, Balaguer, & Aracil, Fundamentos de Robótica, 2007)(Pág. 126)

Una vez obtenida la matriz de transformación homogénea T queda resuelto el problema de la cinemática directa.

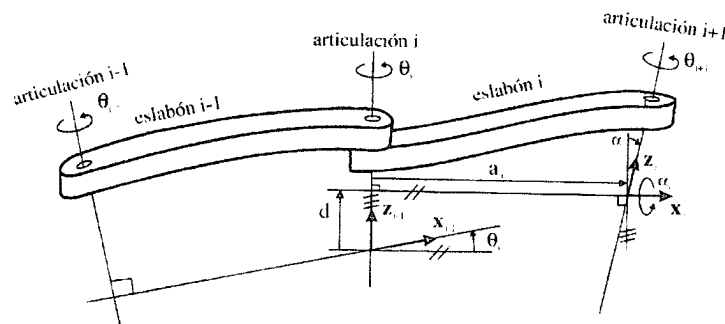


Figura 24 Parámetros Denavit-Hartenberg para eslabón giratorio.

Fuente: (Barrientos, Peñín, Balaguer, & Aracil, Fundamentos de Robótica, 2007, pág. 126)

3.1.6 Cinemática Inversa

En la cinemática directa se busca la posición y orientación del extremo del robot en función de los valores de los parámetros articulares, pero en el trabajo diario de los manipuladores, resulta más práctico hacerlo a la inversa, es decir, a partir de una posición y orientación dada, conseguir los valores articulares que cumplen dicha condición.

Si bien es cierto que en el caso de la cinemática directa la solución se puede conseguir de manera sistemática, en el caso de la cinemática inversa no se puede realizar así. La solución de la cinemática inversa depende mucho de la configuración del robot, y para muchos casos (posiciones y orientaciones) no existe una solución única o simplemente no existe solución. El común para resolver este problema es utilizar métodos numéricos iterativos, sujetos a la velocidad de convergencia y la convergencia misma del resultado; afectando esto el desempeño de los manipuladores, ya que la mayoría de las veces la programación del robot se realiza en tiempo real.

No obstante, a pesar de las dificultades mencionadas, la mayoría de los robots tienen cinemáticas relativamente simples, que facilitan de cierto modo resolución de la cinemática inversa. (Barrientos, Peñín, Balaguer, & Aracil, Fundamentos de Robótica, 1997).

Para la resolución del problema cinemático inverso se puede utilizar el método geométrico, donde se utilizan relaciones geométricas y trigonométricas para conseguir las variables articulares. Otra forma de resolver el problema cinemático es manipulando directamente las ecuaciones obtenidas de la cinemática directa, estableciendo la relación:

$$\begin{bmatrix} n & o & a & p \\ 0 & 0 & 0 & 1 \end{bmatrix} = [t_{ij}] \quad (17)$$

En donde los elementos de t_j son funciones de las variables articulares $[q_1, \dots, q_n]^T$; de la relación de la ecuación (17) se obtienen 12 ecuaciones trigonométricas de donde se pueden despejar n variables articulares q_i en función de los componentes de los vectores n, o, a, p ; pero dicha solución no es fácil de resolver.

Finalmente, se puede dividir el problema cinemático en varias partes con la finalidad de simplificar los cálculos. Generalmente se divide en; posicionamiento y orientación del manipulador; cada uno de éstos se puede resolver de manera independiente y con el método que resulte más fácil. Esta manera de resolver el problema cinemático inverso se le denomina desacople cinemático.

3.1.6.1 Manipulación de la matriz de transformación homogénea para la cinemática inversa.

La finalidad de manipular la matriz de transformación homogénea es conseguir seis ecuaciones de entre el conjunto dado de la relación de la ecuación (17), y formar un sistema de seis ecuaciones con seis incógnitas, para el caso de un robot con seis grados de libertad; que se pueda resolver fácilmente.

El procedimiento para resolver el problema cinemático inverso consiste en: Obtener la matriz T de transformación homogénea que relaciona el sistema de referencia de la base $\{S_0\}$ con el sistema de referencia del extremo $\{S_n\}$, posteriormente y partiendo de que $T = {}^0A_1 {}^1A_2 \dots {}^{n-1}A_n$, despejando se consigue:

$$\begin{aligned} ({}^0A_1)^{-1} T &= {}^1A_2 \dots {}^{n-1}A_n \\ ({}^1A_2)^{-1} ({}^0A_1)^{-1} T &= \dots {}^{n-1}A_n \end{aligned} \quad (18)$$

Y partiendo del hecho que T (Ecuación (17)) es conocida, entonces en la Ecuación (18) los miembros de la izquierda son función de las variables articulares (q_1, \dots, q_k) mientras que los miembros de la derecha son función de las variables articulares (q_{k-1}, \dots, q_n) ; partiendo de este hecho se puede resolver comenzando por resolver q_1 de la primera expresión de la Ecuación (18), seguidamente se puede obtener q_2 de la segunda expresión y así sucesivamente hasta conseguir todas las variables articulares. Para utilizar este método es necesario conocer la matriz inversa de ${}^{i-1}A_i$, que viene dada por:

$$({}^{i-1}A_i)^{-1} = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix}^{-1} = \begin{bmatrix} n_x & n_y & n_z & -n^T p \\ o_x & o_y & o_z & -o^T p \\ a_x & a_y & a_z & -a^T p \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (19)$$

WWW.BDIGITAL.ULA.VE

3.1.6.2 Uso del desacople cinemático

El uso del desacople cinemático se realiza cuando los métodos anteriores se tornan muy complejos en su desarrollo. Este método permite dividir el problema en dos problemas de menor complejidad, un problema de posicionamiento y un problema de orientación; en este último se ha de tener en cuenta que los tres últimos grados de libertad del manipulador, se deben cortar en un mismo punto.

Una vez separados los problemas de posicionamiento y orientación, se parte del punto de resolver el problema del posicionamiento por el método que sea más conveniente, luego a partir de los resultados anteriores, se resuelve el problema de la orientación, buscando las variables articulares, para ello se busca la sub-matriz de rotación, 0R_n , de la matriz de transformación homogénea 0T_n , esta sub-matriz de rotación estará formada por la sub-matriz de rotación del

posicionamiento, 0R_p (donde p es el último sistema de referencia correspondiente a los eslabones de posicionamiento); y la sub-matriz de orientación, pR_n .

$${}^0R_n = [n \quad o \quad a] = {}^0R_p {}^pR_n \quad (20)$$

De la ecuación anterior se puede obtener pR_n :

$${}^pR_n = [r_{ij}] = ({}^0R_p)^{-1} \cdot {}^0R_n = ({}^0R_p)^T [n \quad o \quad a] \quad (21)$$

Siendo el término de la derecha conocido, por lo tanto se pueden obtener los valores de las variables articulares que faltan.

3.1.7 Matriz Jacobiana

Los métodos para la solución de la cinemática directa e inversa, mostrados en los apartados anteriores, se centran únicamente en el posicionamiento y orientación del manipulador robótico desde el punto de vista estático y en ausencia de movimiento del robot. Sin embargo, también es competencia de la cinemática, conocer las relaciones de velocidad existente entre sus componentes, con qué velocidad angular y lineal se mueve el extremo cuando las articulaciones se mueven a una velocidad dada, o viceversa.

Conocer las velocidades angulares y lineales de los componentes del manipulador robótico se realiza a través de un modelo diferencial llamado matriz Jacobiana, y se define como el efecto que un movimiento diferencial de las variables articulares tiene sobre las variables en el espacio de tarea. Esta relación permite que las variables articulares puedan ser manipuladas mediante un sistema de control. En tal sentido, se tiene que el cálculo de la matriz Jacobiana se puede realizar

Licencia Creative Commons:

Atribución - No Comercial - Compartir Igual 3.0 Venezuela
(CC BY-NC-SA 3.0 VE)

a partir de las velocidades de cada articulación para conocer la velocidad del extremo, esta forma de cálculo se llama Jacobiana directa; y calcular las velocidades articulares partiendo de la velocidad de extremo del manipulador se denomina Jacobiana inversa.

En ambos casos, Jacobiana directa e inversa, se puede relacionar las velocidades articulares $(\dot{q}_1, \dot{q}_2, \dots, \dot{q}_n)$ con vectores de velocidades expresados en otro sistema coordenado, por ejemplo, se puede considerar la localización y orientación del extremo del robot en base a sus coordenadas cartesianas y los ángulos de Euler $(\dot{x}, \dot{y}, \dot{z}, \dot{\phi}, \dot{\theta}, \dot{\psi})$, a esta forma de representación se le denomina Jacobiana analítica. También se puede relacionar las velocidades angulares con las velocidades lineales en un mismo sistema de coordenadas $(v_x, v_y, v_z, w_x, w_y, w_z)$, esta relación se obtiene a través de la matriz Jacobiana geométrica.

3.1.7.1 Jacobiana analítica

La Jacobiana analítica relaciona las velocidades articulares $(\dot{q}_1, \dot{q}_2, \dots, \dot{q}_n)$ con las velocidades de localización del extremo del robot $(\dot{x}, \dot{y}, \dot{z}, \dot{\phi}, \dot{\theta}, \dot{\psi})$. Este método consiste en diferenciar las ecuaciones del modelo cinemático directo. Por ejemplo, suponiendo un robot con n grados de libertad se tiene que:

$$\begin{aligned}
 x &= f_x(q_1, \dots, q_n) \\
 y &= f_y(q_1, \dots, q_n) \\
 z &= f_z(q_1, \dots, q_n) \\
 \phi &= f_\phi(q_1, \dots, q_n) \\
 \theta &= f_\theta(q_1, \dots, q_n) \\
 \psi &= f_\psi(q_1, \dots, q_n)
 \end{aligned} \tag{22}$$

Derivando con respecto al tiempo ambos lados de la igualdad del conjunto de ecuaciones anteriores:

$$\begin{aligned}
 \dot{x} &= \sum_1^n \frac{\partial f_x}{\partial q_i} \dot{q}_i \\
 \dot{y} &= \sum_1^n \frac{\partial f_y}{\partial q_i} \dot{q}_i \\
 \dot{z} &= \sum_1^n \frac{\partial f_z}{\partial q_i} \dot{q}_i \\
 \dot{\phi} &= \sum_1^n \frac{\partial f_\phi}{\partial q_i} \dot{q}_i \\
 \dot{\theta} &= \sum_1^n \frac{\partial f_\theta}{\partial q_i} \dot{q}_i \\
 \dot{\psi} &= \sum_1^n \frac{\partial f_\psi}{\partial q_i} \dot{q}_i
 \end{aligned} \tag{23}$$

Simplificando, y expresando en forma matricial:

WWW.BDIGITAL.ULA.VE

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = J_a \cdot \begin{bmatrix} \dot{q}_1 \\ \vdots \\ \dot{q}_n \end{bmatrix} = \begin{bmatrix} \frac{\partial f_x}{\partial q_1} & \dots & \dots & \frac{\partial f_x}{\partial q_n} \\ \vdots & \ddots & & \vdots \\ \vdots & & \ddots & \vdots \\ \frac{\partial f_\psi}{\partial q_1} & \dots & \dots & \frac{\partial f_\psi}{\partial q_n} \end{bmatrix} \cdot \begin{bmatrix} \dot{q}_1 \\ \vdots \\ \dot{q}_n \end{bmatrix} \tag{24}$$

Donde J_a es la matriz Jacobiana analítica, y cada uno de los elementos $[j_{pq}]$ depende de los valores de q_i , razón por la cual la Jacobiana será diferente en cada uno de los puntos del espacio articular.

3.1.7.2 Jacobiana Inversa

Para obtener la velocidad final del extremo del robot partiendo de las velocidades articulares se utiliza la Jacobiana directa. Contrariamente, para calcular los valores de las velocidades articulares partiendo de la velocidad de extremo del manipulador, se utiliza la Jacobiana inversa. Para resolver la Jacobiana inversa se puede utilizar varios métodos, uno de ellos consiste en una vez conocida la Jacobiana directa, invertirla simbólicamente:

$$\begin{bmatrix} \dot{q}_1 \\ \vdots \\ \dot{q}_n \end{bmatrix} = J_a^{-1} \cdot \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \quad (25)$$

Este método, aunque pareciera fácil, en la práctica no lo es. El invertir simbólicamente matrices con funciones trigonométricas es de gran complejidad. Incluso si se realiza la inversión de manera numérica, hay que considerar que la matriz cambia para cada valor de q_i , por lo que su resolución en tiempo real puede que no sea viable; además de que pueden existir ciertas configuraciones del robot para las cuales la matriz Jacobiana inversa no tenga solución o sea nula como en el caso de configuraciones singulares.

Otro método para resolver la matriz Jacobiana inversa, consiste en utilizar el mismo método para conseguir la Jacobiana directa, pero partiendo ahora del modelo cinemático inverso:

$$\begin{aligned} q_1 &= f_1(x, y, z, \phi, \theta, \psi) \\ &\vdots \\ q_n &= f_n(x, y, z, \phi, \theta, \psi) \end{aligned} \quad (26)$$

Diferenciando con respecto al tiempo ambos miembros de la igualdad:

$$\begin{bmatrix} \dot{q}_i \\ \vdots \\ \dot{q}_n \end{bmatrix} = J_a^{-1} \cdot \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} \frac{\partial f_1}{\partial x} & \dots & \dots & \frac{\partial f_1}{\partial \psi} \\ \vdots & & & \\ \vdots & & & \\ \frac{\partial f_n}{\partial x} & \dots & \dots & \frac{\partial f_n}{\partial \psi} \end{bmatrix} \quad (27)$$

3.1.7.3 Configuraciones singulares

Las configuraciones singulares de un manipulador robótico se definen como aquellas configuraciones donde el determinante de la matriz Jacobiana inversa se anula. Al anularse la matriz Jacobiana, un incremento pequeño de las coordenadas cartesianas supone un incremento infinito en las coordenadas articulares, conduciendo a que se produzcan movimientos y/o velocidades inalcanzables por los actuadores. En otras palabras, se pierden grados de libertad, haciendo que sea imposible para el manipulador alcanzar ciertas posiciones en el espacio.

Las singularidades se pueden clasificar en dos, singularidad en la parte baja del espacio de trabajo del robot; cuando está en algún punto del límite de trabajo interno o externo, haciendo que el robot no pueda ir más allá. El segundo tipo de singularidad es en el espacio interior de trabajo, y se produce por la alineación de dos o más ejes de las articulaciones del robot.

Para evitar las configuraciones singulares, debe tenerse en cuenta, desde que se comienza el diseño del manipulador, colocar restricciones de movimiento que los sistemas de control sean capaces de detectar y así evitar estas posiciones.

3.1.8 Dinámica del robot

La dinámica estudia el movimiento de los cuerpos debido a fuerzas externas e internas de los mismos. Desde el punto de vista de la robótica, se estudia la relación entre el movimiento de los eslabones y las fuerzas o pares ejercidos por los actuadores en las articulaciones. Una vez estudiada la dinámica del robot, se puede controlar a cada uno de los actuadores en las articulaciones, para que el extremo del robot se mueva de una manera determinada.

De igual manera que en el caso de la cinemática, en la dinámica surgen dos problemas, el de la dinámica directa y el de la dinámica inversa. En ambos casos se hace uso de la matemática para relacionar la localización del robot, sus derivadas, las fuerzas y pares aplicados en las articulaciones y los parámetros dimensionales del robot. Específicamente, en la dinámica directa se calcula el movimiento del extremo partiendo de las fuerzas y pares en los actuadores; mientras que en la dinámica inversa se calculan las fuerzas y pares en los actuadores para que el extremo del manipulador realice un movimiento determinado.

Las ecuaciones obtenidas para la resolución del problema dinámico se pueden resolver de varias maneras, por métodos iterativos para resolver la dinámica de forma numérica; por medio de soluciones cerradas, en las que se plantean soluciones para un tipo de robot específico, pero se tienen que repetir muchos cálculos; y finalmente de forma recursiva, que son las más adecuadas para resolver los problemas en tiempo real (Torres, Pomares, Gil, Puente, & Aracil, 2002).

Evidentemente la solución al problema de la dinámica no es fácil, y es uno de los aspectos más complejos en la robótica, tanto así que suele ser obviado en numerosas ocasiones (Barrientos, Peñín, Balaguer, & Aracil, Fundamentos de Robótica, 1997), sin embargo, resolver el problema puede ser esencial para: realizar simulaciones de movimiento del robot; diseño y evaluación de la estructura mecánica; dimensionamiento de los actuadores; y diseño y evaluación del control dinámico del robot.

Aunque es posible estudiar las leyes de la dinámica con diferentes principios, Newton, D'Alembert, ecuaciones de Lagrange, principios de Hamilton; en robótica se emplean

principalmente dos principios. El primero es la formulación Lagrangiana, que establece un balance de energías a través del lagrangiano. El segundo principio es la formulación de Newton-Euler, que realiza un balance de pares y fuerzas existentes. Este último principio permite una implementación computacional recursiva de manera más fácil (Torres, Pomares, Gil, Puente, & Aracil, 2002).

3.1.8.1 Modelo dinámico – Newton-Euler

La formulación de Newton-Euler se basa en la formulación de ecuaciones que describen el movimiento lineal y angular de los eslabones, y las fuerzas que intervienen. Este método parte del equilibrio de fuerzas para cada eslabón:

$$\begin{aligned}\sum F_i &= \frac{d}{dt}(m_i v_i) = m_i \dot{v}_i \\ \sum T_i &= \frac{d}{dt}(I_i \omega_i) = I_i \dot{\omega}_i + \omega_i \times (I_i \omega_i)\end{aligned}\tag{28}$$

Donde:

F_i son las fuerzas ejercidas sobre la barra i

T_i son los pares ejercidos sobre la barra i en torno a su centro de masa

m_i es la masa de la barra i

I_i es el tensor de inercia de la barra i en torno a su centro de masas, expresado en el sistema de referencia $\{S_i\}$

$v_i - \dot{v}_i$ son la velocidad y aceleración lineal del centro de masa de la articulación i

$\omega_i - \dot{\omega}_i$ son la velocidad y aceleración angular de la articulación i

Partiendo del desarrollo de las ecuaciones (28), y utilizando una formulación recursiva, se obtienen la posición, velocidad y aceleración del eslabón i referido a la base del robot. Para ello se utiliza un algoritmo descrito por Barrientos (2007), basado en la formulación Newton-Euler, que se basa en operaciones vectoriales y matriciales, que reducen el nivel de complejidad

computacional de los cálculos, llevándolo a un nivel $O(n)$, lo que quiere decir que depende directamente del número de eslabones. El algoritmo basado en la formulación Newton-Euler para el desarrollo dinámico de un robot es el siguiente:

1. Asignar a cada eslabón un sistema de referencia de acuerdo a las normas de D-H
2. Establecer las condiciones iniciales:

Para el sistema de la base $\{S_0\}$:

$${}^0\omega_0 : \text{velocidad angular} = [0, 0, 0]^T$$

$${}^0\dot{\omega}_0 : \text{aceleración angular} = [0, 0, 0]^T$$

$${}^0v_0 : \text{velocidad lineal} = [0, 0, 0]^T$$

$${}^0\dot{v}_0 : \text{aceleración lineal} = -[g_{x0}, g_{y0}, g_{z0}]^T$$

${}^0\omega_0, {}^0\dot{\omega}_0, {}^0v_0$, son típicamente nulos salvo que la base del robot está en movimiento.

$[g_{x0}, g_{y0}, g_{z0}]$ es el vector de gravedad expresado en el sistema $\{S_0\}$ (habitualmente toma el valor $[0, 0, -9.8]$ pues z_0 se sitúa vertical hacia arriba.

Para el extremo del robot se conocerá la fuerza y el par ejercidos externamente ${}^{n+1}f_{n+1}$ y ${}^{n+1}n_{n+1}$

$$z_0 = [0, 0, 1]^T$$

${}^i p_i$ = Vector que une el origen $\{S_{i-1}\}$ con el de $\{S_i\}$ expresadas en $\{S_i\} = [a_i, d_i \sin(\alpha_i), d_i \cos(\alpha_i)]$

${}^i s_i$ = Coordenadas del centro de masa del eslabón i respecto del sistema $\{S_i\}$

${}^i I_i$ = Matriz de inercia del eslabón i expresado en un sistema paralelo al $\{S_i\}$ y con el origen en el centro de masa del eslabón

3. Obtener las matrices de rotación ${}^{i-1}R_i$ y sus inversas ${}^i R_{i-1} = ({}^{i-1}R_i)^{-1} = ({}^{i-1}R_i)^T$,

$${}^{i-1}R_i = \begin{bmatrix} \cos \theta_i & -\cos \alpha_i \sin \theta_i & \sin \alpha_i \sin \theta_i \\ \sin \theta_i & \cos \alpha_i \cos \theta_i & -\sin \alpha_i \cos \theta_i \\ 0 & \sin \alpha_i & \cos \alpha_i \end{bmatrix}$$

Para $i = 1 \dots n$ realizar los pasos 4 a 7.

4. Obtener la velocidad angular del sistema $\{S_i\}$

$${}^i \omega_i = \begin{cases} {}^i R_{i-1} ({}^{i-1} \omega_{i-1} + z_0 \dot{q}_i) & \text{si el eslabón } i \text{ es de rotación} \\ {}^i R_{i-1} {}^{i-1} \omega_{i-1} & \text{si el eslabón } i \text{ es de traslación} \end{cases} \quad (29)$$

5. Obtener la aceleración angular del sistema $\{S_i\}$

$${}^i \dot{\omega}_i = \begin{cases} {}^i R_{i-1} ({}^{i-1} \dot{\omega}_{i-1} + z_0 \ddot{q}_i) + {}^{i-1} \omega_{i-1} \times z_0 \dot{q}_i & \text{si el eslabón } i \text{ es de rotación} \\ {}^i R_{i-1} {}^{i-1} \dot{\omega}_{i-1} & \text{si el eslabón } i \text{ es de traslación} \end{cases} \quad (30)$$

6. Obtener la aceleración lineal del sistema i :

$${}^i \dot{v}_i = \begin{cases} {}^i \omega_i \times {}^i p_i + {}^i \omega_i \times ({}^i \omega_i \times {}^i p_i) + {}^i R_{i-1} {}^{i-1} \dot{v}_{i-1} & \text{Rotación} \\ {}^i R_{i-1} (z_0 \ddot{q}_i + {}^{i-1} \dot{v}_{i-1}) + {}^i \omega_i \times {}^i p_i + 2 {}^i \omega_i \times {}^i R_{i-1} z_0 \dot{q}_i + {}^i \omega_i \times ({}^i \omega_i \times {}^i p_i) & \text{Traslación} \end{cases} \quad (31)$$

7. Obtener la aceleración lineal del centro de gravedad del eslabón i :

$${}^i a_i = {}^i \omega_i \times {}^i s_i + {}^i \omega_i \times ({}^i \omega_i \times {}^i s_i) + {}^i \dot{v}_i \quad (32)$$

Para $i = n \dots 1$ realizar los pasos 8 a 10

8. Obtener la fuerza ejercida sobre el eslabón i :

$${}^i f_i = {}^i R_{i+1} {}^{i+1} f_{i+1} + m_i {}^i a_i \quad (33)$$

9. Obtener el par ejercido sobre el eslabón i :

$${}^i n_i = {}^i R_{i+1} [{}^{i+1} n_{i+1} + ({}^{i+1} R_i {}^i p_i) \times {}^{i+1} f_{i+1}] + ({}^i p_i + {}^i s_i) \times m_i {}^i a_i + {}^i I_i {}^i \omega_i + {}^i \omega_i \times ({}^i I_i {}^i \omega_i) \quad (34)$$

10. Obtener la fuerza o par aplicado a la articulación i :

$$\tau = \begin{cases} {}^i n_i^{T_i} R_{i-1} z_0 & \text{si el eslabón } i \text{ es de rotación} \\ {}^i f_i^{T_i} R_{i-1} z_0 & \text{si el eslabón } i \text{ es de traslación} \end{cases} \quad (35)$$

Donde τ es el par o fuerza efectivo (para motor menos pares de rozamiento o perturbación).

(Barrientos, Peñín, Balaguer, & Aracil, Fundamentos de Robótica, 2007, págs. 232-233)

3.2 Herramientas computacionales para el diseño de robots manipuladores

En la actualidad, el uso de herramientas computacionales para casi cualquier área del conocimiento está muy difundido, y el diseño de robots no escapa a esta realidad. Son muchas las herramientas computacionales existentes en el mercado utilizadas para diseñar robots, en este caso en específico, robots manipuladores. Existen herramientas matemáticas generales como el software MATLAB® de la casa Mathworks; Maple® de la casa de software MapleSoft. También hay herramientas especializadas que operan bajo los programas mencionados anteriormente como “Robotic Toolbox for MATLAB” desarrollado por Peter Corke. Adicional a las herramientas computacionales matemáticas, existen los programas CAD (Diseño Asistido por Computadora) de uso general, que permiten el diseño geométrico y mecánico de los manipuladores, estos programas pueden modelar tridimensionalmente y realizar simulaciones. Entre este tipo de herramientas encontramos Autodesk Inventor® y CATIA® desarrollado por la casa de software Dassault Systems.

Adicional a las herramientas computacionales genéricas, se tienen herramientas computacionales específicas para el diseño y simulación de robots; muchas de esta son de licencia de código abierto, mientras que otras son de uso propietario. Entre las herramientas especializadas se

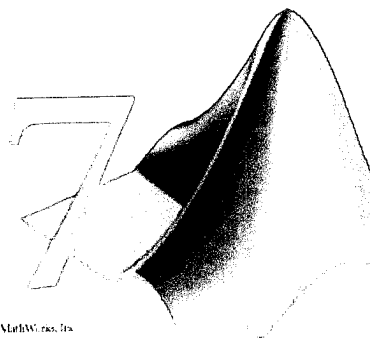
pueden encontrar programas de simulación como Robolab II, RoboSIM, JRoboOP entre otros. Una descripción de las características más resaltantes se puede encontrar en (Arismendi, Aguirre, & Barrios, 2008).

3.2.1 MATLAB ®

MATLAB es una herramienta computacional utilizada para realizar cálculos matemáticos. Realizado por la casa de software MathWorks, es considerado un lenguaje de alto nivel capaz de realizar cálculos complejos de forma más rápida que los lenguajes tradicionales como C, C++, Fortran (MathWorks, 2007).

Al ser una herramienta de tipo genérico, se puede utilizar para una gran variedad de tareas como: cálculo técnico, procesamiento de señales, diseño de sistemas de control, modelado de sistemas, etc. Entre sus características principales se encuentran: Gestión de código, archivos y datos; Funciones matemáticas de álgebra, trigonometría, estadística, optimización, etc. (MathWorks, 2007).

MATLAB®



Copyright © 1984-2006, The MathWorks, Inc.

Figura 25 MATLAB 7.
Fuente: MathWorks

3.2.2 MAPLE ®

MAPLE ® es una herramienta computacional realizada por MapleSoft, y especializada en el cálculo simbólico. Es capaz de realizar cálculos matemáticos geométricos, trigonométricos; también puede resolver modelado de sistemas; simulaciones; optimizaciones, etc. Entre sus principales características se puede mencionar que cubre casi todas las áreas de la matemática, realiza gráficas y animaciones en 2D y 3D, genera código directo para C, C++, Fortran, MATLAB ®; por último tiene conexión directa con Excel ®, MATLAB ®, Fortran y otros (MapleSoft, 2007).

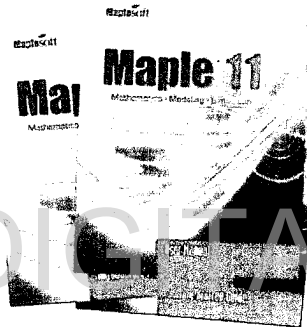


Figura 26 Maple 11.
Fuente: Maplesoft

3.2.3 Robotic Toolbox for MATLAB ®

Robotic Toolbox for MATLAB ®, son un conjunto de herramientas específicas para el diseño y simulación de manipuladores robóticos, que funciona bajo el programa MATLAB ®. Este conjunto de herramientas provee soluciones específicas para las cinemáticas, dinámicas, Jacobianas y generación de trayectorias. Trabaja con la notación de Denavit-Hartenberg y con la notación D-H modificada (Craig, 2006), e implementa el algoritmo recursivo de Newton-Euler. (Corke, 2006)

3.2.4 Autodesk Inventor ®

Es un modelador de sólidos paramétricos 3D, es producido por la casa de software Autodesk ®. Es un programa CAD utilizado principalmente para el diseño mecánico. Funciona principalmente creando piezas por separado, para luego ser ensambladas; de allí se puede probar y simular dichos montajes. (Wikipedia, 2008)

Expresando lo que dice la empresa Autodesk ® de su software en la página web oficial:

Autodesk ® Inventor es un software que es la base para el prototipado digital. El modelo 3D de Inventor es un prototipo digital 3D preciso que permite al usuario validar los datos de diseño e ingeniería al mismo tiempo que se trabaja, minimiza la necesidad de los prototipos físicos, y reduce los costos de ingeniería debido a cambios una vez que el diseño ha sido enviado a manufactura. Con el acercamiento tecnológico para acelerar y simplificar el proceso desde el concepto hasta la manufactura, Inventor ha sobrepasado a todos sus competidores por seis años consecutivos. (Autodesk, 2008)



Figura 27 Autodesk Inventor 2008.
Fuente: Autodesk Inc.

Capítulo 4:

Brazo Manipulador propuesto (Prototipo)

En el presente capítulo se muestra el proceso de diseño del manipulador robótico, el diseño geométrico, el cálculo de la cinemática directa e inversa, las matrices Jacobianas, el cálculo de la dinámica del manipulador, selección de actuadores y sensores. Se muestran también los resultados obtenidos.

Se comienza con el diseño mecánico geométrico del manipulador, donde se muestra los materiales utilizados, dimensiones seleccionadas, actuadores, sensores, elementos de transmisión de energía y movimiento entre otros. El siguiente apartado muestra el cálculo de las cinemáticas directa e inversa, con lo que se definen los movimientos del manipulador. Se calcula la matriz Jacobiana a partir de la cinemática y seguidamente se hace referencia a la dinámica del manipulador.

Una vez definido el manipulador, se realizan comparaciones entre los resultados del cálculo realizado y los resultados obtenidos con la herramienta para el cálculo y simulación de robots “Robotic toolbox for MATLAB”. Al finalizar con la comparación de los resultados, se muestra el desarrollo de un programa computacional para la simulación y futura programación del brazo manipulador propuesto. En el último apartado de este capítulo, se realiza el control del brazo manipulador propuesto.

4.1 Diseño Mecánico

El manipulador robótico diseñado posee tres grados de libertad, que le permiten un posicionamiento en un espacio tridimensional. Los grados de libertad son articulaciones del tipo rotacional, que serán accionados por motores paso a paso. El movimiento será supervisado por sensores que midan posición angular, en este caso encoders y/o potenciómetros. El material de fabricación será aluminio por ser un material liviano y de fácil de manejo en los procesos de fabricación.

4.1.1 Diseño detallado

Para llegar al diseño del manipulador robótico, primero se realizó una revisión de algunas de las principales casas fabricantes de manipuladores robóticos; se buscó específicamente los modelos más pequeños, ya que por su tamaño son ideales para la formación, entrenamiento e investigación. Una vez revisados se realiza una tabla para su comparación, ver Tabla 2, en ella se muestra el alcance máximo del manipulador, los tamaños de los eslabones. En la Figura 28 se muestra una imagen referencial donde se muestran los eslabones.

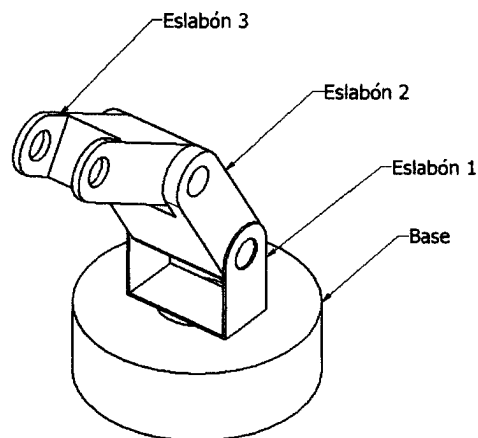


Figura 28 Eslabones de los manipuladores

Tabla 2. Robots Manipuladores Comerciales, dimensiones en mm.

Robot Manipulador	Alcance	Eslabón 1	Eslabón 2	Eslabón 3
FANUC LR Mate 200iB	700	250	290	80
FANUC LR Mate 100iB	620	250	220	80
FANUC LR Mate 200iB/3L	858	330	370	80
KUKA KR3	635	265	270	75
ABB IRB 140	810	360	380	65
ADEPT Adept Viper 850	854	365	405	80

Luego de la revisión de los manipuladores comerciales, se nota que a pesar de ser los más pequeños de cada una de las casas seleccionadas, siguen siendo grandes para el propósito del presente trabajo, por lo tanto, se realizan propuestas de tamaños de robots manipuladores, con dimensiones más pequeñas, tomando en cuenta de la Tabla 2 que la mayoría de los manipuladores tienen el eslabón 2 mayor que el eslabón 1, y que el eslabón 3 es el más pequeño de todos. Presentar propuestas más pequeñas permite que el manipulador a diseñar tenga mayor capacidad para el transporte del mismo, pudiéndose llevar para las aulas de clases, laboratorios, congresos, presentaciones, etc. y no tener la necesidad de un espacio adaptado sólo para el uso del mismo. En la Tabla 3 se muestran las propuestas realizadas para el manipulador robótico en estudio.

Tabla 3 Propuestas de tamaño para la propuesta de manipulador. Dimensiones en mm.

Propuesta	Alcance	Eslabón 1	Eslabón 2	Eslabón 3
Propuesta 1	630	250	300	80
Propuesta 2	480	150	200	130
Propuesta 3	335	180	190	65
Propuesta 4	425	200	165	60

Al tener las propuestas, se tomaron la Propuesta 3 y la Propuesta 4 para un análisis, pero luego de revisar el comportamiento cinemático de las mismas se decide descartarlas. Posteriormente se

revisa la Propuesta 2, igualmente se revisa el comportamiento cinemático de la misma, y esta vez se decide trabajar con esta propuesta. Los cálculos y resultados de la cinemática se muestran en el apartado dedicado a la misma, más adelante.

El dimensionamiento de la base del brazo manipulador, en un principio se basa sólo en la altura de la misma, y se toma en consideración para la misma, el tamaño de los motores a utilizar como elementos actuadores. Se toma entonces una medida de 80 mm de altura, y las demás dimensiones se definirán posteriormente. La Figura 29 muestra un esquema con las medidas generales del manipulador.

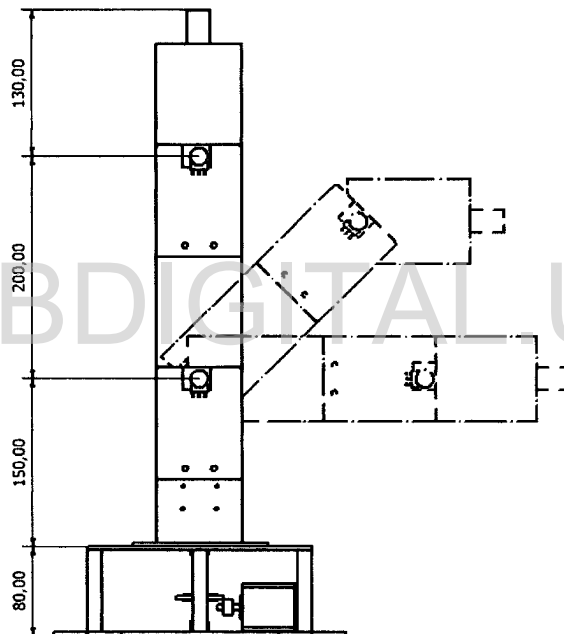


Figura 29 Medidas generales del manipulador

Una vez definidas las dimensiones generales, se evalúan los posibles materiales a utilizar junto con los posibles fabricantes de la carcasa que pueden intervenir en el proceso de fabricación, también los actuadores, sensores y mecanismos de transmisión entre otros.

Los principales factores a tomar en cuenta para la selección de los materiales son costos, disponibilidad en el país, peso y mano de obra local capacitada para su manufactura. Para el material se decide trabajar con metal por su rigidez, resistencia y durabilidad, adoptando el de aluminio.

Basados en los criterios de selección para el material, se observó que la diferencia entre el acero y el aluminio no es mucha. Ambos materiales se consiguen en el mercado, así como la mano de obra calificada para su manufactura. Finalmente la diferencia de peso entre el aluminio y el acero es notable, siendo la densidad del acero (en general) mayor que la del aluminio. Por esta razón principalmente se decide trabajar con aluminio como material para el diseño del brazo manipulador.

Una vez seleccionado el material, se buscan alternativas para la forma de los eslabones, así como las medidas faltantes para el dimensionamiento total de los mismos. Se tiene como primera propuesta eslabones con sección en "U", y con agujeros en las caras para disminuir aún más el peso de los mismos. Los eslabones 1 y 3 tendrían exactamente las mismas dimensiones del perfil, mientras que el eslabón 2 tendría el perfil un poco más pequeño que los otros eslabones, de manera que se pudieran acoplar y permitir el movimiento rotacional de las articulaciones. Ver Figura 30.

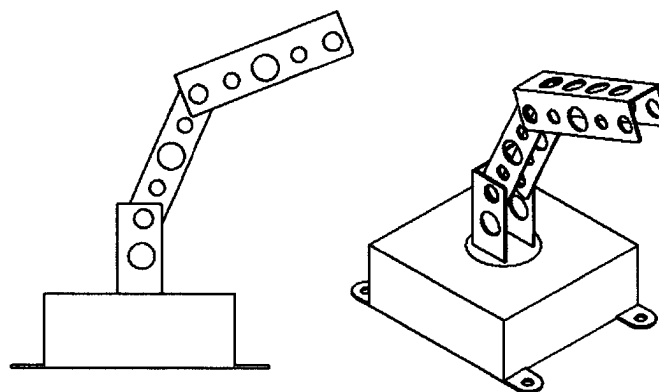


Figura 30 Primera propuesta, perfil en "U"

La primera propuesta no se toma en cuenta por varias razones, los eslabones tienen los movimientos articulares restringidos, esto sucede cuando los ejes longitudinales de los eslabones se alinean, en este caso sólo se admiten movimientos hacia adelante y no hacia atrás, y en muchos casos, dependiendo del tipo de trabajo es necesario conseguir este tipo de movimientos. Otra razón es que los agujeros por ser tan grandes no dejan el espacio suficiente para el acoplamiento de los demás elementos como motores y sensores. Finalmente el perfil en “U” puede resultar no tan rígido cuando el brazo manipulador realice movimientos laterales, por lo que sería recomendable colocar refuerzos.

Se realiza una segunda propuesta, ver Figura 31, el perfil es completamente cerrado, de forma cuadrada, y para simplificar más aún la fabricación y el diseño, se utiliza un perfil comercial cuadrado de aluminio de 76,2 mm (3 pulgadas) de lado y 2 mm de espesor. Luego a los eslabones 1 y 2 se les colocan extensiones hechas de chapa de aluminio de 3 mm de espesor, para crear los puntos de unión entre los eslabones.

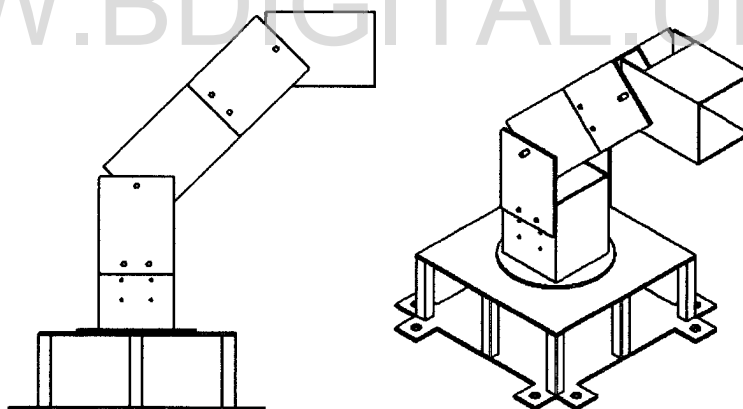


Figura 31 Segunda propuestas, perfil cuadrado.

La segunda propuesta presenta movimientos articulares más completos, permitiendo movimientos más amplios, hacia adelante y hacia atrás, cubriendo un recorrido de aproximadamente 210 grados. Los eslabones son más rígidos, y tienen más área para la sujeción

de otros elementos como motores y sensores. Estas razones llevan a tomar la decisión de trabajar con esta estructura para el diseño y fabricación del brazo manipulador.

Cada uno de los eslabones es diferente en cuanto a su configuración y tamaño. Para identificar cada uno de los eslabones, se comienza con el eslabón número uno (1) que es el que está unido a la base, el eslabón número dos (2) es el que está unido al número uno, y finalmente el eslabón número tres (3) es el final de la cadena cinemática, está unido al eslabón número dos (2).

El eslabón número uno consta de tres partes, una base circular de 120 mm de diámetro y 3 mm de espesor, donde estará fijado el eslabón como tal, y con la que por medio de un eje se fijará a la base del manipulador. El cuerpo del eslabón está formado por perfil cuadrado de aluminio de 76,20 mm (3 pulgadas) de lado, y una altura de 97 mm. Finalmente al cuerpo del eslabón están fijadas dos extensiones rectangulares de 96 x 72,20 mm (96 mm x 3") y 3 mm de espesor; la extensión tiene tres agujeros, uno en la parte superior donde va el eje que une el eslabón número uno con el eslabón número dos, y dos agujeros en la parte inferior que sirven para realizar soldadura de tapón y fijar mejor la extensión al cuerpo del eslabón. Ver Figura 32 y Figura 33.

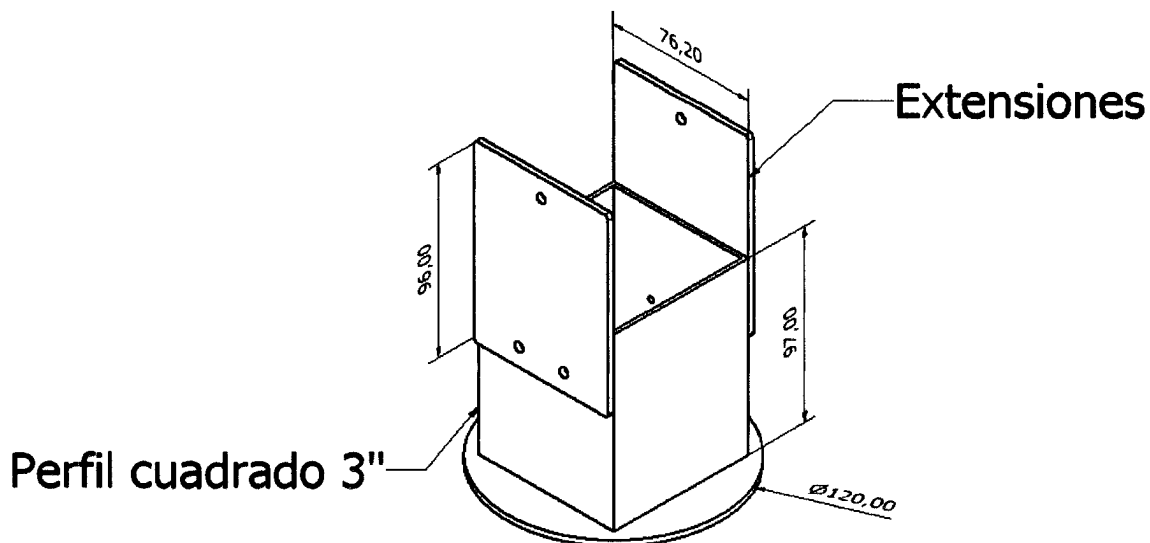


Figura 32 Eslabón número uno

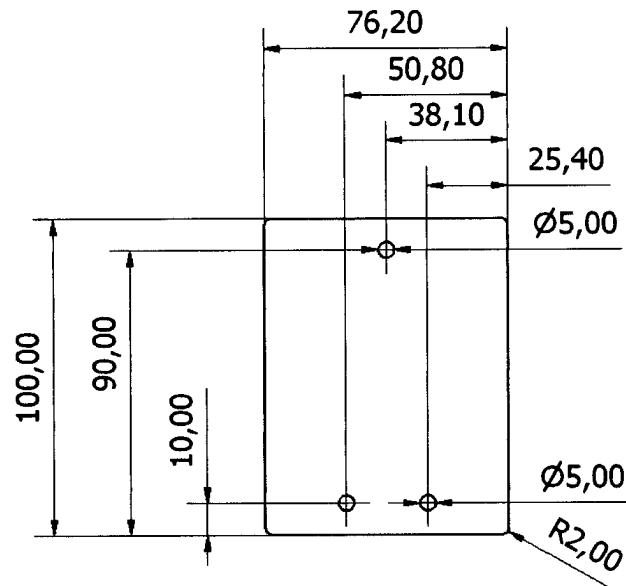


Figura 33 Extensiones

El eslabón número dos, ver Figura 34, está compuesto de dos partes, el cuerpo formado de perfil cuadrado de aluminio de 76,20 mm (3 pulgadas) de lado tiene una altura de 160 mm, en la parte inferior en dos de las caras se quitó un pedazo de pared de 30 mm, con la finalidad de mejorar el movimiento entre eslabones, y evitar interferencias en caso de colocar componentes dentro del eslabón como los motores, sensores o mecanismos de transferencia de movimiento. En la parte inferior, en las dos caras que quedaron completas se abren dos agujeros por donde pasa el eje que une el eslabón número uno con el eslabón número dos. La otra parte por la que está compuesto el eslabón número dos, que va soldado en la parte superior del cuerpo, es exactamente igual a las extensiones del eslabón número uno, en este caso el agujero superior de la extensión es por donde se unen el eslabón número dos con el eslabón número tres a través de un eje.

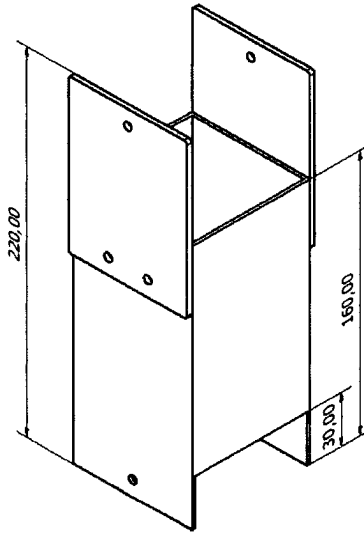


Figura 34 Eslabón número dos

El eslabón número tres, al ser el último eslabón, está formado por perfil cuadrado de 76,20 mm (3 pulgadas), en la parte inferior en dos de las caras se quitó un pedazo de pared de 30 mm, con la finalidad de mejorar el movimiento entre eslabones, y evitar interferencias en caso de colocar componentes dentro del eslabón como los motores, sensores o mecanismos de transferencia de movimiento. En la parte inferior, en las dos caras que quedaron completas se abren dos agujeros por donde pasa el eje que une el eslabón número dos con el eslabón número tres. Ver Figura 35.

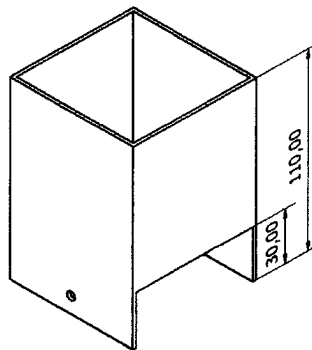


Figura 35 Eslabón número tres

La unión entre los eslabones número uno y número dos se realiza por medio de un eje de 5 mm de diámetro, 120 mm de longitud y de material aluminio. El eje se inserta en los agujeros de la parte inferior del eslabón dos, y en la parte superior del eslabón uno, luego se suelda el eje al eslabón número dos. De igual manera para la unión del eslabón número dos y el eslabón número tres, en este caso el eje va soldado al eslabón número tres, Figura 37.

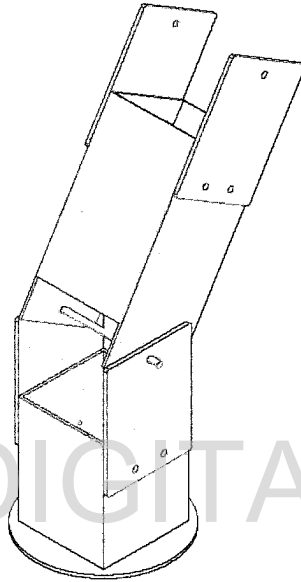


Figura 36 Unión eslabones uno y dos

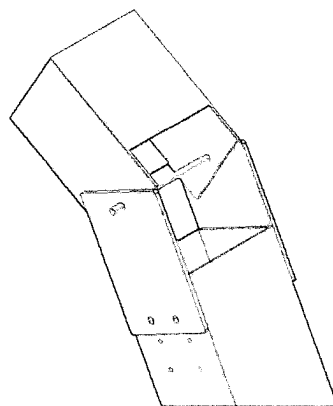


Figura 37 Unión eslabones dos y tres

Luego se construye la base del manipulador, compuesta por dos tapas de chapa metálica de 3 mm de espesor, cuadradas de 200 mm de lado, ver Figura 38. La tapa de inferior tiene dos orejas en cada lado, con agujeros de 10 mm de diámetro, por donde la base puede ser fijada a la superficie de trabajo. La tapa superior tiene un agujero de 5 mm de diámetro por donde pasa el eje que une el eslabón número uno con la base. Ambas tapas de la base están unidas entre sí por ocho tubos de aluminio de 6,35 mm (1/4 pulgada) de diámetro y 74 mm de altura, soldados a las tapas; dejando la base con una altura total de 80 mm. Ver Figura 39.

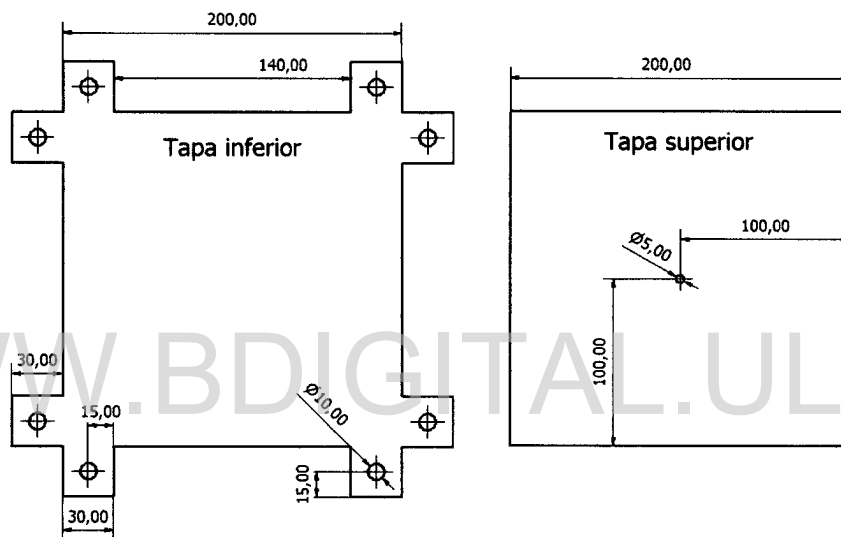


Figura 38 Tapas superior e inferior de la base

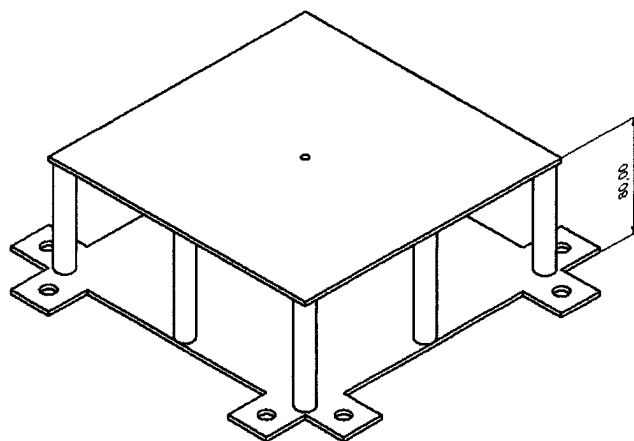


Figura 39 Base del Manipulador

Al eslabón número uno se suelda un eje de aluminio de 5 mm de diámetro y 85 mm de longitud, por medio del cual será unido a la base del manipulador. Una vez soldado el eje al eslabón número uno, se introduce por el agujero de la tapa superior del eslabón. El eje será fijado a la base por medio de dos rodamientos SKF 625-2Z, fijos a la tapa superior e inferior, que permitirán el movimiento rotacional de los eslabones minimizando el roce. Estos rodamientos serán colocados cada uno en una carcasa, que serán las que se fijen mediante soldadura a la tapa superior e inferior. Ver Figura 40.

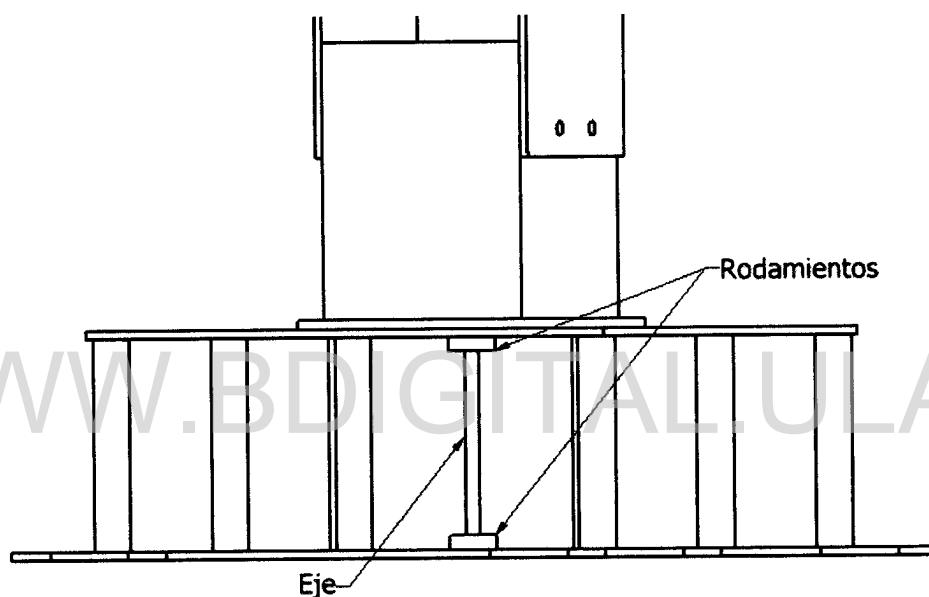


Figura 40 Unión de la base con el eslabón número uno

A medida que se diseñan las piezas del manipulador, se van digitalizando tridimensionalmente en el programa CAD Autodesk Inventor®, de manera de hacer análisis y comprobaciones del mecanismo mientras se diseña, y así minimizar errores durante la fase de diseño. Una vez digitalizadas cada una de las piezas de los eslabones, se realiza el ensamblaje de las mismas, y se comprueba hasta donde se puede realizar el movimiento de las mismas. Ver Figura 41, Figura 42 y Figura 43.

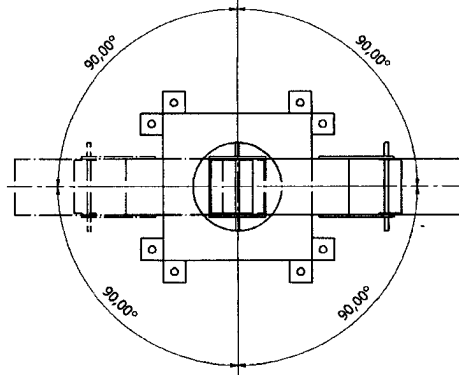


Figura 41 Movimientos eslabón uno. Vista superior

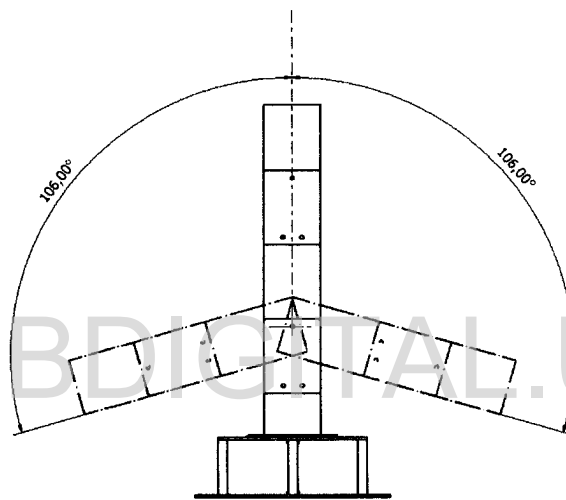


Figura 42 Movimientos eslabón dos

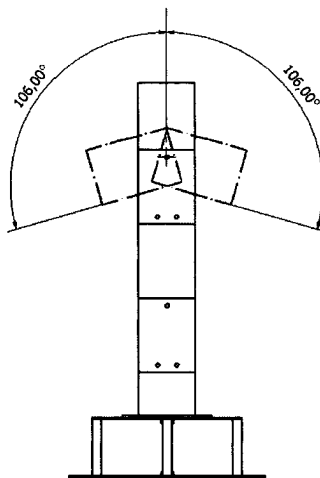


Figura 43 Movimientos eslabón tres

Licencia Creative Commons:
Atribución - No Comercial - Compartir Igual 3.0 Venezuela
(CC BY-NC-SA 3.0 VE)

De la Figura 41 se observa que el eslabón número uno puede realizar un movimiento circular de 360 grados, es decir, tiene total libertad de movimiento. La Figura 42 muestra que los movimientos del eslabón número dos son más limitados, y tienen un rango de 212 grados. Por último, en la Figura 43 se observa que el eslabón número tres también tiene al campo de acción limitado, con un rango de 212 grados.

A partir también del modelo digital, se obtiene el peso de la estructura del manipulador (eslabones, base y ejes), que es de 1,632 Kg. En este caso se suponen todas las piezas de aluminio 6061, que tiene una densidad de $2,710 \text{ g/cm}^3$.

4.1.2 Accionadores.

Los accionadores son los que transforman las señales provenientes del controlador en movimiento de las articulaciones. Por lo tanto, son los encargados del movimiento del manipulador. Los accionadores utilizados en la robótica de manipuladores son básicamente los utilizados en otro tipo de máquinas, pero son tres tipos de accionadores los más utilizados para generar el movimiento. Los tipos de accionadores se pueden clasificar según el tipo de accionamiento que utilizan y sus características, por lo tanto se consiguen del tipo eléctrico, neumático e hidráulico. La selección del tipo de accionador a utilizar viene en función de varias variables como lo son: tamaño del manipulador, tipo de trabajo a utilizar, ambiente de trabajo del manipulador, requerimientos de velocidad y peso a manejar, etc. Según Torres (2002) los más utilizados son los de tipo eléctrico. En la Tabla 4 se muestran las ventajas y desventajas de los tipos de accionadores.

Tabla 4 Principales ventajas y desventajas de los distintos tipos de accionadores.

Tipo de accionador	Ventajas	Desventajas
Eléctrico	<ul style="list-style-type: none"> - Rápidos y precisos. - Posibilidad de aplicar varias técnicas de control del movimiento. - Más económicos. - Tamaño reducido. - Tiempos de respuesta rápidos. 	<ul style="list-style-type: none"> - Altas velocidades implican bajo par. - Necesidad de engranajes o transmisiones. - El juego de los engranajes limita la precisión. - No resultan adecuados en atmósferas inflamables.

		<ul style="list-style-type: none"> - Sobrecaentamiento en condiciones de trabajo de parada. - Necesidad de frenos para bloquear el sistema. - Coste alto en motores grandes.
Neumático	<ul style="list-style-type: none"> - Más económicos. - Alta velocidad. - No contamina el área de trabajo con otros fluidos. - No necesita línea de retorno la instalación. - Fuente de energía usual en entornos industriales. 	<ul style="list-style-type: none"> - Compresibilidad del aire limita el control y la precisión. - Mala precisión al actuar con cargas. - Necesidad de instalación adicional.
Hidráulico	<ul style="list-style-type: none"> - Relación potencia – peso muy buena. - Muy buen servocontrol. - Autolubricación y autorrefrigerado. - Trabajo en parada sin problemas. - Respuesta rápida. - Operación suave a bajas velocidades. - Adecuado en atmósferas inflamables. 	<ul style="list-style-type: none"> - Instalación hidráulica costosa. - Necesidad de mantenimiento y fugas de aceite. - Necesidad de línea de retorno en la instalación. - Problemas de miniaturización.

Fuente: (Torres, Pomares, Gil, Puente, & Aracil, 2002, pág. 50)

Tomando en cuenta la Tabla 4, y la finalidad del manipulador del presente trabajo, se decide utilizar motores eléctricos. Al ser un manipulador robótico con fines educativos, se pueden implementar distintas técnicas de control para demostraciones a los estudiantes, además se puede realizar investigación de nuevas técnicas de control. Adicionalmente son más económicos, y no necesitan instalaciones adicionales especiales para su funcionamiento.

Dentro de los accionadores eléctricos se pueden encontrar varios tipos de motores, motores de corriente alterna, motores de corriente continua, motores paso a paso y servomotores. Se decide trabajar con motores paso a paso, que son motores accionados por un tren de impulsos eléctricos, que hacen mover al rotor de manera discreta, es decir, gira un determinado ángulo por cada paso. Este ángulo de giro puede llegar a ser muy pequeño, por lo que se puede obtener una alta precisión y repetibilidad en cuanto al posicionamiento (Wikipedia, 2006).

Para seleccionar el motor paso a paso a utilizar, se buscan distintos modelos de manera de poder seleccionar dependiendo de las características que ofrecen. A continuación en la Tabla 5 se muestran los motores buscados.

Tabla 5 Motores paso a paso para la selección

Modelo	Compañía	Características
535-0366	RS Automation	Torque: 12 N.cm 200 pasos – 1,8° por paso. Voltaje de alimentación: 12 V Dimensiones: 28x28x51 mm
535-0350	RS Automation	Torque: 9.5 N.cm 200 pasos – 1,8° por paso. Voltaje de alimentación: 12 V Dimensiones: 28x28x45 mm
535-0530	RS Automation	Torque: 126 N.cm 200 pasos – 1,8° por paso Voltaje de alimentación: 22 – 60 V Dimensiones: 65x65x100 mm
CO-5718X-01P	LIN Engineering	Torque: 0,71 N.m Pasos: 400; 800; 1600; 3200; 6400; 12800; 25600; 51200 Voltaje de alimentación: 12 – 40 V Dimensiones: 57x47x86,6 mm
CO-5718L-01P	LIN Engineering	Torque: 2 N.m Pasos: 400; 800; 1600; 3200; 6400; 12800; 25600; 51200 Voltaje de alimentación: 12 – 40 V Dimensiones: 57x47x121,4 mm
DO-4118M-03P	LIN Engineering	Torque: 0,21 N.m Pasos: 200; 400; 800; 1600 Voltaje de Alimentación: 12 – 24 V Dimensiones: 42x42x68,3 mm Encoder integrado
DO-4118M-03P	LIN Engineering	Torque: 0,32 N.m Pasos: 200; 400; 800; 1600 Voltaje de alimentación: 12 – 24 V Dimensiones: 42x42x74,2 mm Encoder integrado

De la Tabla 5, se selecciona el motor modelo DO-4118M-03P por ser un motor pequeño, tener bastante precisión y además tiene el sensor para la medición del movimiento, el encoder. El motor seleccionado no pudo ser conseguido localmente ni nacionalmente, razón por la que se trató de comprar en el exterior, pero razones de tipo económico que sufre el país, no se pudo

Licencia Creative Commons:

Atribución - No Comercial - Compartir Igual 3.0 Venezuela
(CC BY-NC-SA 3.0 VE)

importar el mismo. Se opta por buscar local y nacionalmente los motores paso a paso que se puedan conseguir. Se consiguen dos motores paso a paso, y sus características se muestran en la Tabla 6.

Tabla 6 Motores paso a paso conseguidos nacionalmente

Motor	Características
Motor A	Fabricante: Desconocido Torque: Desconocido Voltaje de alimentación: 12 V Pasos: 225 (1,6°) Dimensiones: 42x42x46 mm
Motor B	Fabricante: Desconocido Torque: Desconocido Voltaje de alimentación: 5 V Pasos: 200 (1,8°) Dimensiones: ϕ 42 x 23 mm

Se prueban ambos motores sobre el modelo físico del manipulador, pero ninguno de los motores tuvo la fuerza suficiente para mover los eslabones, razón por la cual no son tomados en cuenta. Finalmente se tiene como opción utilizar servomotores que vienen incluidos en un kit de robótica de LEGO®, específicamente Lego Mindstorms NXT® ítem #8725, y que pertenece al autor.

Los servomotores son motores de corriente continua, que tienen la capacidad de colocarse en cualquier posición dentro del rango de operación, y mantenerse estables en la misma. Están compuestos de un motor paso a paso, con un conjunto de engranajes reductores que se encargan de convertir el giro en torsión. (Wikipedia, 2006). Los servomotores LEGO® seleccionados tienen las características que se muestran en la Tabla 7. La Figura 44 muestra el motor a utilizar, y la Figura 45 muestra como está compuesto el motor internamente.

Tabla 7 Motor LEGO® NXT

Motor	características
Motor NXT. Ítem# 9842 LEGO®	Peso: 80 g Voltaje de alimentación: 9 V Velocidad de rotación: 170 rpm Torque: 50 N.cm (Estacionario) Tren de engranajes de reducción. Sensor para movimiento rotacional, encoder.

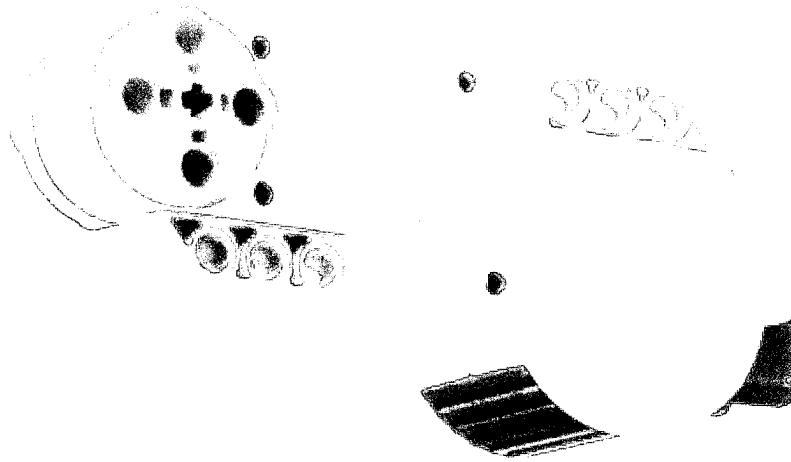


Figura 44 Motor LEGO NXT.
Fuente: (LEGO (R), 2006)

WWW.BDIGITAL.ULA.VE

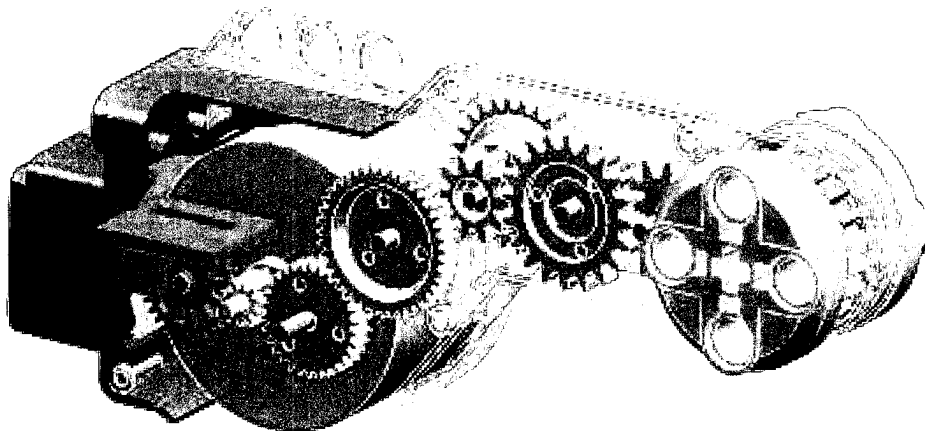


Figura 45 Motor LEGO NXT - Tren de engranes de reducción.
Fuente: (LEGO Education, 2006)

Licencia Creative Commons:
Atribución - No Comercial - Compartir Igual 3.0 Venezuela
(CC BY-NC-SA 3.0 VE)

4.1.3 Sensores

Los sensores utilizados en el manipulador robótico en estudio, son para la medición de la posición angular de las articulaciones. Para este tipo de medición se pueden utilizar varios tipos de sensores, entre los que se encuentran los encoders, resolvers y potenciómetros. A continuación, en la Tabla 8, se muestra una comparación entre estos tres tipos de sensores.

Tabla 8 Comparación entre distintos sensores de posición angular.

Sensor	Robustez mecánica	Rango dinámico	Resolución	Estabilidad térmica
Encoder	Mala	Media	Buena	Buena
Resolver	Buena	Buena	Buena	Buena
Potenciómetro	Regular	Mala	Mala	Mala

Fuente: (Barrientos, Peñín, Balaguer, & Aracil, Fundamentos de Robótica, 2007, pág. 55)

En un principio, y tomando en cuenta la comparación de la Tabla 8 se busca utilizar un sensor de tipo resolver, estos sensores emiten una señal analógica de la medición tomada, por lo que es necesario utilizar un convertidor analógico digital para la señal; mientras que los encoders emiten la señal digital directamente. Por esta razón se decide cambiar y utilizar un encoder en vez de un resolver.

Al seleccionar el primer motor a utilizar, se escogió uno que tiene integrado el sensor, para ese caso un encoder de 400 ciclos por revolución y 1600 pulsos; pero por las razones mencionadas no se utilizó dicho motor. Al no poder seleccionar el primer motor y decidir utilizar motores con disponibilidad nacional, se emprende la búsqueda de sensores tipo encoders a nivel local y nacional, siendo esta infructuosa. Por este motivo se decide utilizar potenciómetros, que son comercializables en el país. Finalmente al quedar descartados los motores encontrados a nivel nacional y utilizar motores LEGO ® NXT que tienen incluidos encoders, se decide no tomar en cuenta los potenciómetros.

Los encoders utilizados por los motores NXT de LEGO ® son encoders ópticos y tienen precisión de +/- 1 grado, y reconocen el sentido de giro. Se tiene la ventaja de que viene acoplado al motor, por lo que no se requiere de componentes adicionales.

4.1.4 Transmisión de movimiento y potencia.

El movimiento a realizar por las articulaciones es rotacional, al igual que el de los motores, por lo que el tipo de transmisión es rotacional-rotacional. Para realizar este tipo de transmisión son varios los sistemas que se pueden utilizar, pero los más comunes, con sus ventajas y desventajas se muestran en la Tabla 9.

Tabla 9 Sistemas de transmisión

Denominación	Ventajas	Desventajas
Engranajes	Pares altos	Holguras
Correa dentada	Distancias grandes	Deslizamientos
Cadena	Distancias grandes	Ruido
Cable	Distancias grandes	Deformabilidad

Fuente: (Torres, Pomares, Gil, Puente, & Aracil, 2002)

Los motores seleccionados ya cuentan con un tren de engranajes de reducción, razón por la que se pueden utilizar minimizando el uso de reducciones adicionales. Para el diseño del manipulador del presente trabajo se utilizan engranajes acoplados directamente al motor y al eje de movimiento de las articulaciones. Los engranajes utilizados son plásticos, también del kit de robótica de LEGO ®, se utiliza uno grande para el eje de la articulación, 40 dientes y 41,7 mm de diámetro, y uno pequeño acoplado al motor, 8 dientes y 9,3 mm de diámetro, ver Figura 46; ambos engranajes tienen un espesor de diente de 3,8 mm. Con estos engranajes se consigue una relación de 5:1, es decir que por cada cinco vueltas que gira el engranaje del motor, el engranaje del eje gira una vuelta, consiguiendo un mayor control y precisión sobre el movimiento de la articulación.

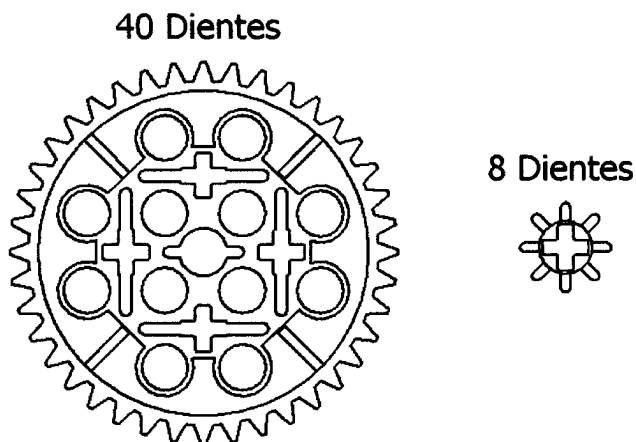


Figura 46 Engranajes de LEGO seleccionados

Los engranajes de LEGO® inicialmente vienen en el centro con un agujero en forma de cruz, como se muestra en el engranaje pequeño de la Figura 46; pero el engranaje grande, de 40 dientes, fue modificado, abriéndole en el centro un agujero de 5 mm, para poder acoplarlo al eje de la articulación, también de 5 mm. El engranaje pequeño no sufre modificaciones, ya que será acoplado al motor con un eje de sección transversal en forma de cruz, también sacado del kit de robótica de LEGO®.

4.1.5 Ensamblaje del robot manipulador propuesto

Con los eslabones y la base ya ensamblados, y seleccionados los actuadores y sistemas de transmisión de potencia, se termina de ensamblar el robot manipulador, primero se realiza un prototipo virtual para minimizar y/o eliminar previamente los errores que puedan existir; y luego se realiza el ensamblaje físicamente, completando así el prototipo físico.

Se comienza el ensamblaje con los eslabones número dos y número tres. Se parte de la premisa que los engranajes para la transmisión de potencia y movimiento se encuentren bien alineados, y la línea que une los centros de los engranajes sea perpendicular al eje que une los eslabones. Ver Figura 47.

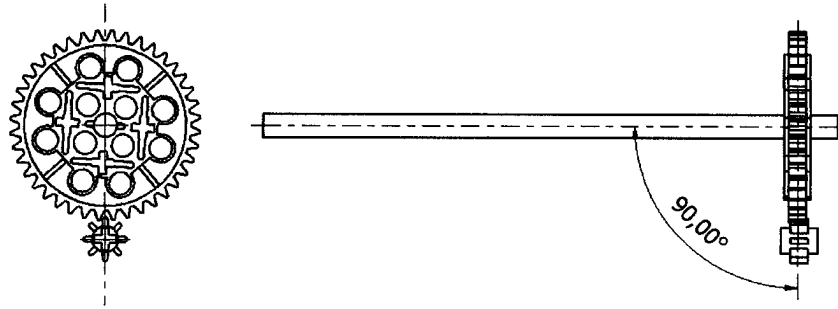


Figura 47 Alineación de los engranajes

Para lograr la alineación de los engranajes, se toman otras piezas del mismo kit de robótica de LEGO®, en este caso las columnas Technic®, que por ser piezas que pertenecen al mismo kit de robótica LEGO®, permitirán que el acople y la alineación sean los más adecuados; estas piezas vienen de diferentes tamaños, diferenciándose por el número de agujeros que trae la columna, para el caso del manipulador en estudio, se utilizan sólo tres tamaños de columnas: #5; #7 y #9. Ver Figura 48.

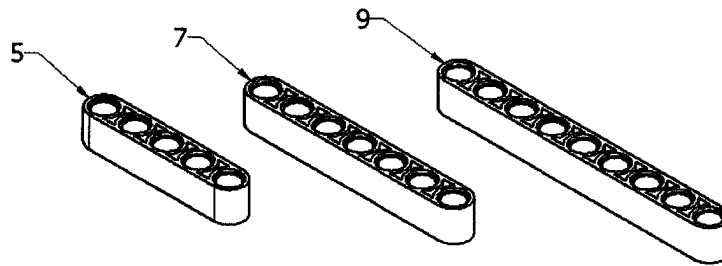


Figura 48 Columnas Technic® de LEGO®

Se toman dos columnas #9 para fijarlas al eslabón número dos, una para cada lado, de manera que el eje que une los eslabones número dos y número tres entre uno de los agujeros extremos de la columna de LEGO®, se asegura que la columna quede vertical completamente, para luego utilizar los tres agujeros inferiores uno por medio y marcar el eslabón número dos para abrir

agujeros de 3,5 mm de diámetro, por donde pasarán tornillos de 3 mm de diámetro que unirán la columna con el eslabón. Ver Figura 49.

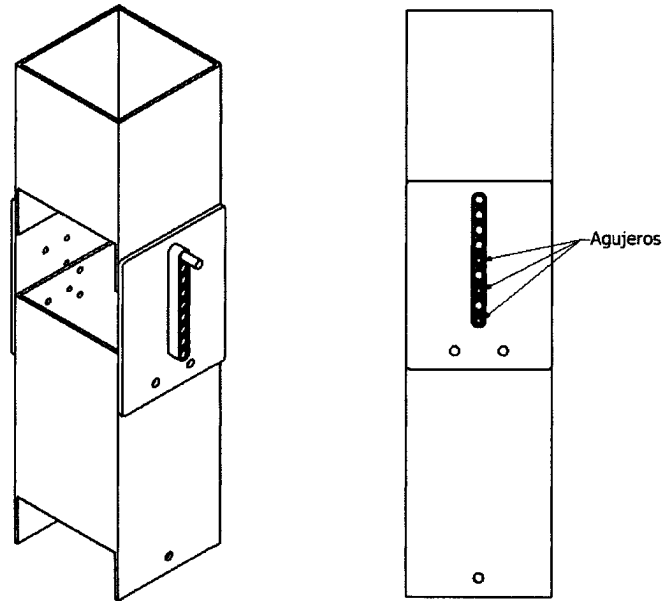


Figura 49 Columnas LEGO #9 para la unión de los eslabones dos y tres

Una vez abiertos los agujeros, se utilizan tornillos avellanados AS 1427 M3x20, para unir las columnas con el eslabón número dos, del otro lado, por la parte interna del eslabón, los tornillos se sujetan con tuercas AS 1474 M3. Ver Figura 50.

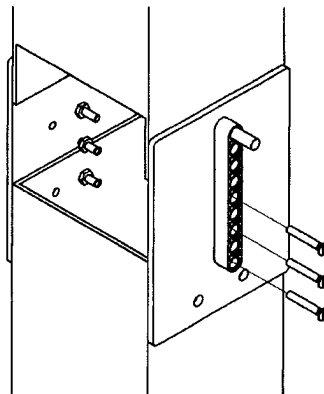


Figura 50 Unión de las columnas LEGO con el eslabón tres

Una vez sujeta las columnas LEGO #9 con el eslabón número dos, se coloca el engranaje de 40 dientes sobre el eje que une los eslabones dos y tres. Adicionalmente se colocan en uno de los lados del eslabón número tres, del mismo lado que se colocó el engranaje; dos columnas LEGO ® de donde se va a sujetar el motor LEGO ® NXT ® que moverá los engranajes, se coloca una columna #5 y una columna #7. Cada una de las columnas se sujeta con dos tornillos AS 1427 M3x20 y con tuercas AS 1474 M3. Ver Figura 51.

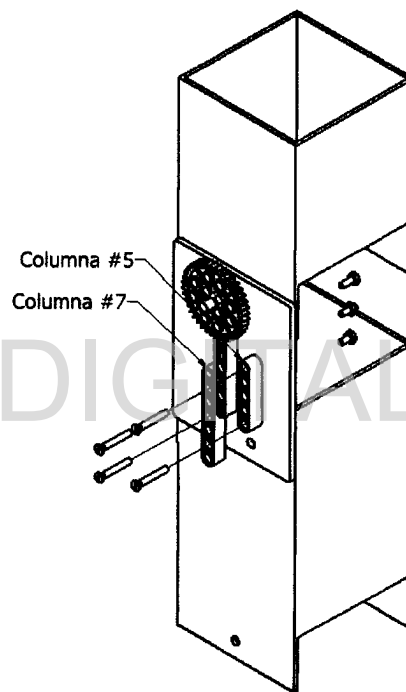


Figura 51 Unión columnas LEGO #5 y #7 al eslabón dos

Posteriormente se ensambla el motor LEGO ® NXT ® que será colocado en el eslabón número dos, de manera que pueda mover el engranaje de 40 dientes colocado sobre el eje que une los eslabones número dos y número tres. Para ello, se utilizan piezas de unión también del kit de robótica NXT de la compañía LEGO ®, eje de sección en forma de cruz #5 que tiene una longitud de 39 mm, pines de cuatro puntas, pines sencillos de 24 mm de longitud y de 16 mm longitud. Ver Figura 52

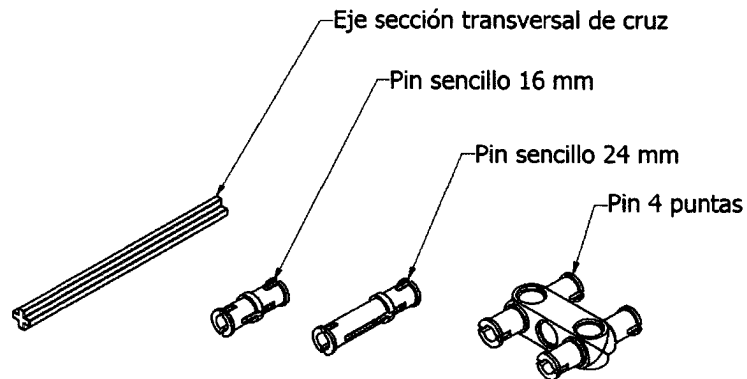


Figura 52 Elementos de unión LEGO utilizados

Se inserta el eje #5 en el cabezal del motor, luego el engranaje de 8 dientes se inserta también en el eje, hasta que haga tope con el cabezal del motor. Posteriormente se insertan un pin de 4 puntas en los agujeros superiores del motor, y un pin de 4 puntas en los agujeros de la parte inferior del motor. Seguidamente se inserta una columna #5 en el pin de 4 puntas que se encuentra en la parte inferior del motor. Por último, se insertan dos pines sencillos, uno de 16 mm y uno de 24 mm en la columna #5. Este ensamblaje será montado sobre las columnas que se encuentran en el eslabón número dos. Ver Figura 53 y Figura 54.

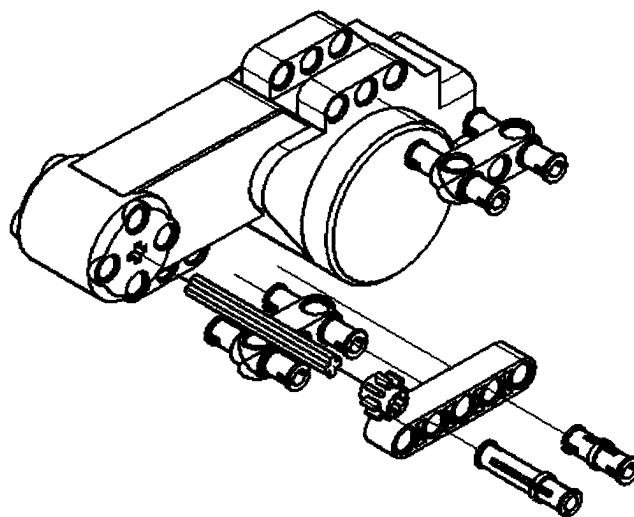


Figura 53 Ensamblaje de motor para eslabones

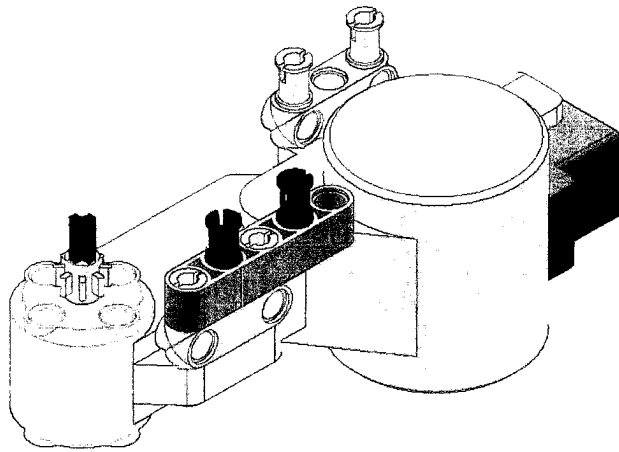


Figura 54 Motor ensamblado para eslabones

Una vez ensamblado el motor, el mismo se inserta en las columnas que se encuentran en las paredes del eslabón número dos. De esta manera queda completado el ensamblaje entre los eslabones número dos y número tres. Ver Figura 55 y Figura 56.

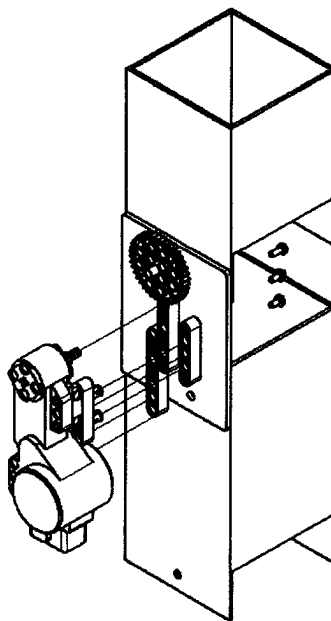


Figura 55 Ensamblaje del motor con eslabón dos

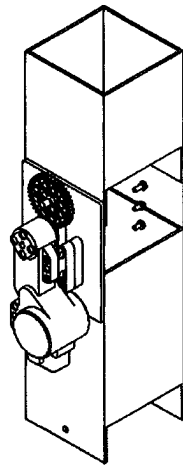


Figura 56 Motor ensamblado en eslabón dos

Terminado el ensamblaje del eslabón número dos con el eslabón número tres, se procede a ensamblar el eslabón número uno con el eslabón número dos. Para este ensamblaje se realizan los mismos pasos que para el ensamblaje de los eslabones dos y tres, con la única diferencia que esta vez se utilizan dos motores para el ensamblaje. Estos motores se ensamblan de manera que al rotar lo hagan en el mismo sentido. Ver Figura 57.

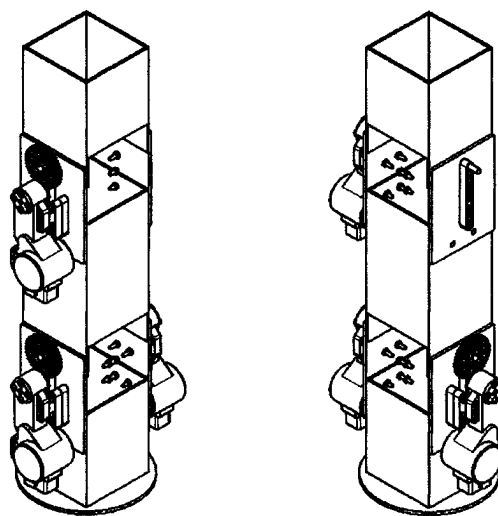


Figura 57 Motores ensamblados al eslabón uno

El uso de dos motores en el ensamblaje de los eslabones uno y dos, corresponde al hecho que este movimiento de la articulación dos necesita mayor potencia, debido a que cuando el eslabón dos y tres se encuentran en la posición inicial, es cuando producen mayor momento en la articulación número dos. Pruebas con un solo motor fueron realizadas para mover el eslabón dos, donde se observó que un solo motor si puede mover el eslabón, pero lo hace muy lento aún y cuando el motor está a máxima capacidad; adicionalmente se observó que el motor trabaja de manera forzada. Luego se realizaron pruebas con dos motores, donde se observó que el movimiento del eslabón se realizaba más rápido, y los motores no trabajaban de manera forzada, ya que el torque que deben vencer se reparte entre los mismos.

Ensamblados completamente los eslabones uno y dos, se procede a ensamblar la base del manipulador con el eslabón número uno, para así de esta manera completar el ensamblaje del manipulador robótico en estudio.

Se comienza colocando el engranaje de 40 dientes en el eje que une el eslabón número uno con la base. Luego se alinea el engranaje de 8 dientes unido al motor, de manera que el eje que une los engranajes sea perpendicular al eje del motor, ver Figura 58. En esta oportunidad se difiere de la manera como el motor se ensambló con los demás eslabones, debido al espacio reducido que se tiene en la base; encontrándose que esta manera de colocar el motor es bastante cómoda para su posterior manipulación y conexión con los cables de transmisión de energía y datos.

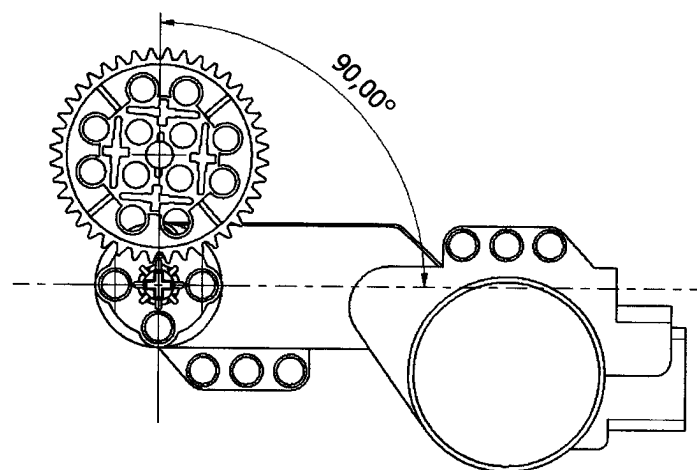


Figura 58 Alineación engranajes con motor de la base

Después de haber colocado el motor, se miden sobre la tapa inferior de la base los agujeros que hay que hacer para colocar las columnas LEGO® Technic® de donde se va a sostener el motor. Para sujetar las columnas se utilizan también tornillos avellanados de 3 mm de diámetro, para ello se realizan agujeros avellanados desde el fondo de la base, de manera que al colocar el tornillo la cabeza quede completamente alineada con la base de la tabla y no afecte la estabilidad de la base. Ver Figura 59.

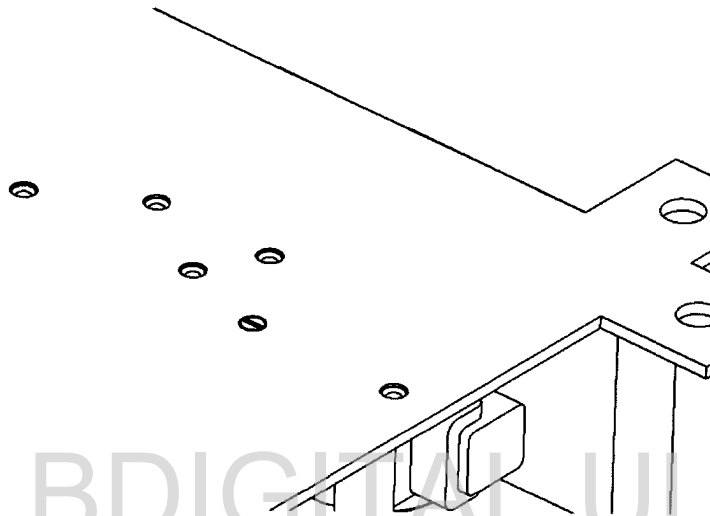


Figura 59 Agujeros fondo de la base

Una vez hechos los agujeros, se colocan por el lado interno de la tapa inferior de la base las columnas de LEGO®, se coloca una columna #9 y una #7 para sostener el motor, y se coloca una columna #7 alineada con el eje de rotación del motor, que sirve para guiar el engranaje pequeño de 8 dientes, y también para evitar que el eje que sostiene el engranaje sufra pandeo. Las columnas se fijan con tornillos avellanados AS 1427 M3x20 y con tornillos AS 1427 M3x12 donde no había espacio para los tornillos más largos. Para terminar la sujeción se utilizan tuercas AS 1474 M3. Ver Figura 60.

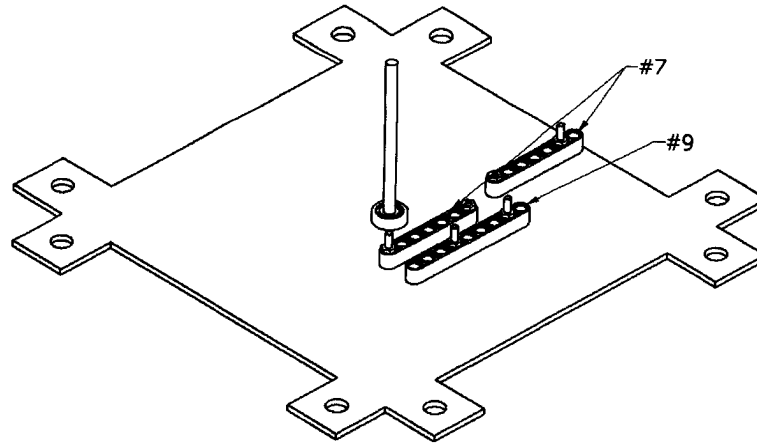


Figura 60 Columnas LEGO para sostener el motor en la base

En la tapa superior de la base se realiza el mismo procedimiento anterior, pero solamente se coloca una columna LEGO ® #5 alineada con el eje donde se encuentra el engranaje de 8 dientes, de manera de sostenerlo y evitar que el mismo se desalinee o sufra pandeo que afecte la transmisión entre los engranajes. Ver Figura 61.

WWW.BDIGITAL.ULA.VE

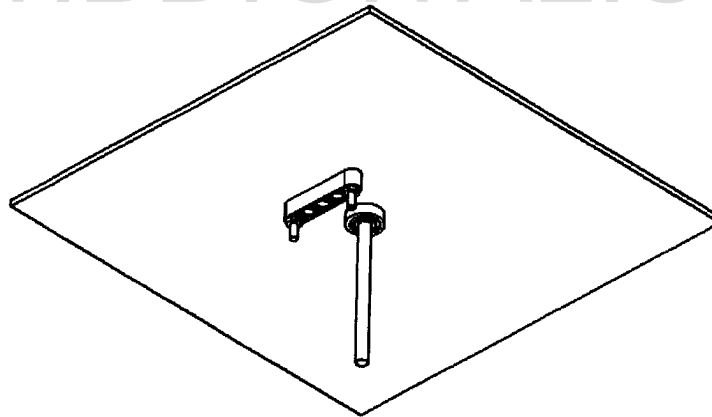


Figura 61 Columna #5 en la tapa superior de la base

Se ensambla el motor LEGO ® NXT ® con los conectores, para ello se utilizan dos pines sencillos de 16 mm que son colocados en los agujeros superiores del motor, un conector de 24 pines en los agujeros inferiores del motor. Luego se inserta un eje #9 en el motor, sobre el eje se

Licencia Creative Commons:
Atribución - No Comercial - Compartir Igual 3.0 Venezuela
(CC BY-NC-SA 3.0 VE)

inserta un separador de 8 mm, y sobre el separador se inserta el engranaje de 8 dientes, completando así el ensamblaje del motor a utilizar en la base. Ver Figura 61 y Figura 62.

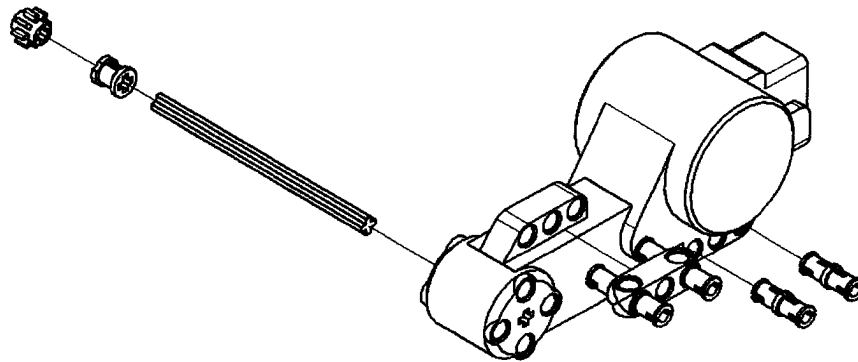


Figura 62 Despiece motor para la base

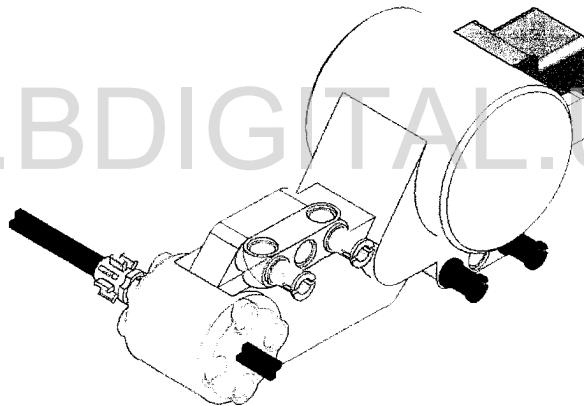


Figura 63 Motor de la base ensamblado

Una vez ensamblado el motor se inserta en las columnas Technic® que se encuentran en la base, completado así el ensamblaje de la base y el ensamblaje del manipulador robótico en estudio. Ver Figura 64 y Figura 65 respectivamente. En la Figura 66 y la Figura 67 se puede observar el prototipo del robot manipulador propuesto ya fabricado. El peso total del manipulador ensamblado es de 2 Kg aproximadamente.

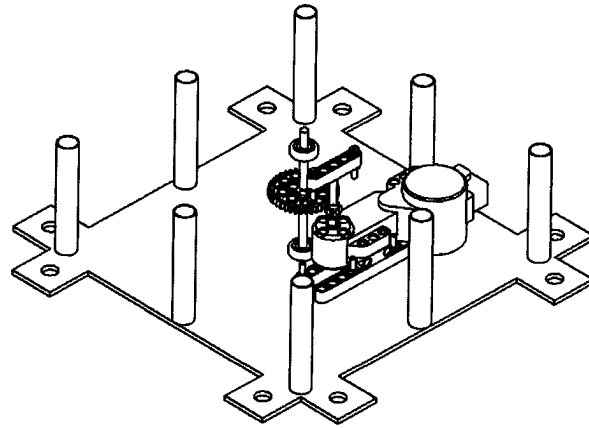


Figura 64 Base ensamblada

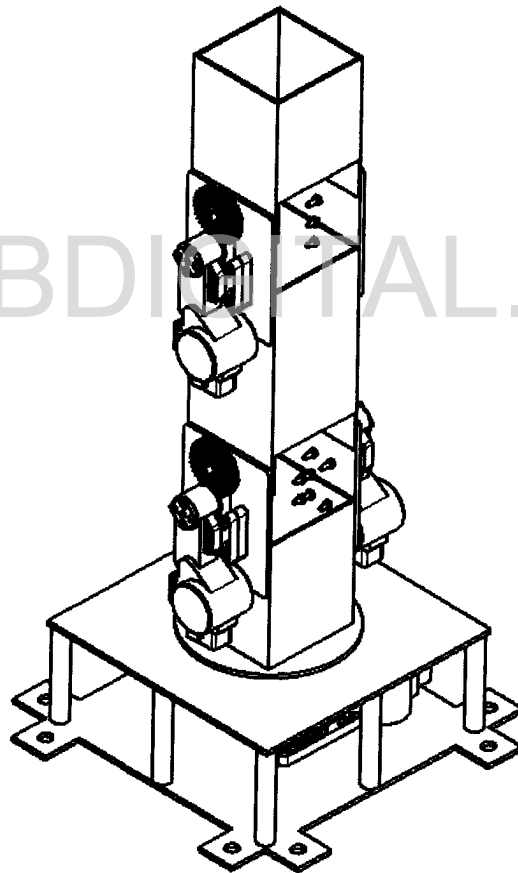


Figura 65 Manipulador ensamblado

Licencia Creative Commons:
Atribución - No Comercial - Compartir Igual 3.0 Venezuela
(CC BY-NC-SA 3.0 VE)

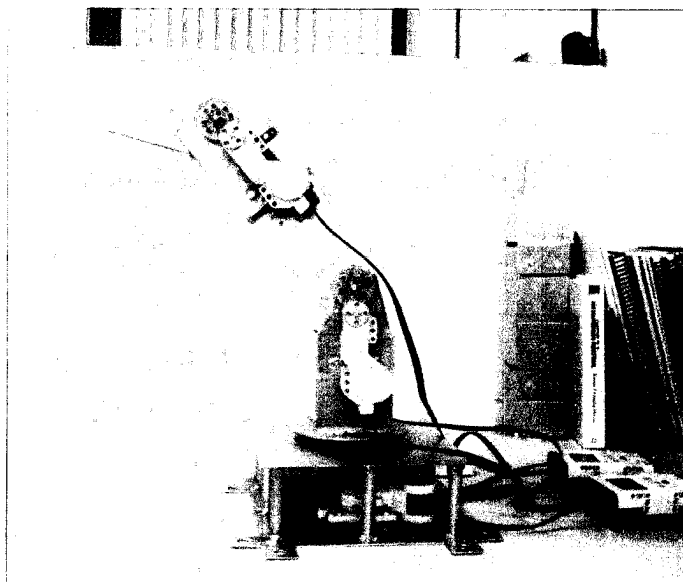


Figura 66 Prototipo del robot manipulador propuesto

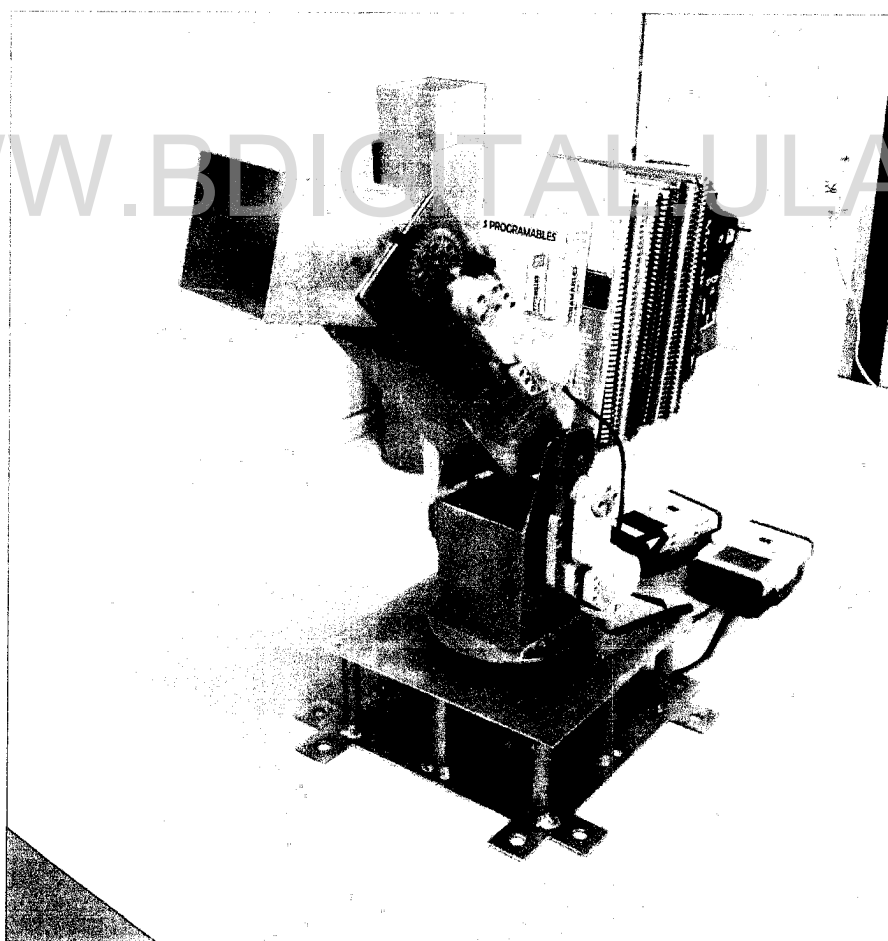


Figura 67 Prototipo del robot manipulador propuesto

Licencia Creative Commons:
Atribución - No Comercial - Compartir Igual 3.0 Venezuela
(CC BY-NC-SA 3.0 VE)

4.2 Cinemática del manipulador propuesto

En este apartado se presentan las cinemáticas directa e inversa del manipulador propuesto, así como las matrices Jacobianas del mismo. El cálculo se realiza con las matrices homogéneas, por lo que se necesita de los parámetros Denavit-Hartenberg. Para conseguir estos parámetros es necesario primero definir el sistema de coordenadas del manipulador y de cada una de las articulaciones. Luego siguiendo el algoritmo de Denavit-Hartenberg, se numeran los eslabones, comenzando con el eslabón número cero que es la base, luego los eslabones uno, dos y tres. Las articulaciones se numeran como uno la articulación entre el eslabón cero y el uno; articulación dos la que une los eslabones uno y dos; finalmente la articulación tres la que une los eslabones dos y tres. Ver Figura 68 y Figura 69

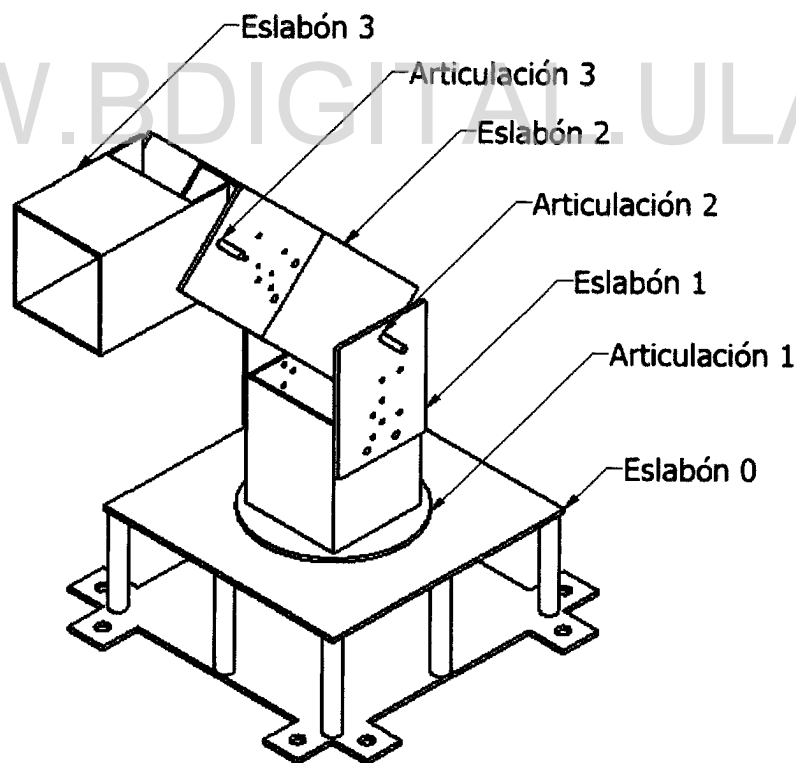


Figura 68 Identificación de los eslabones y las articulaciones

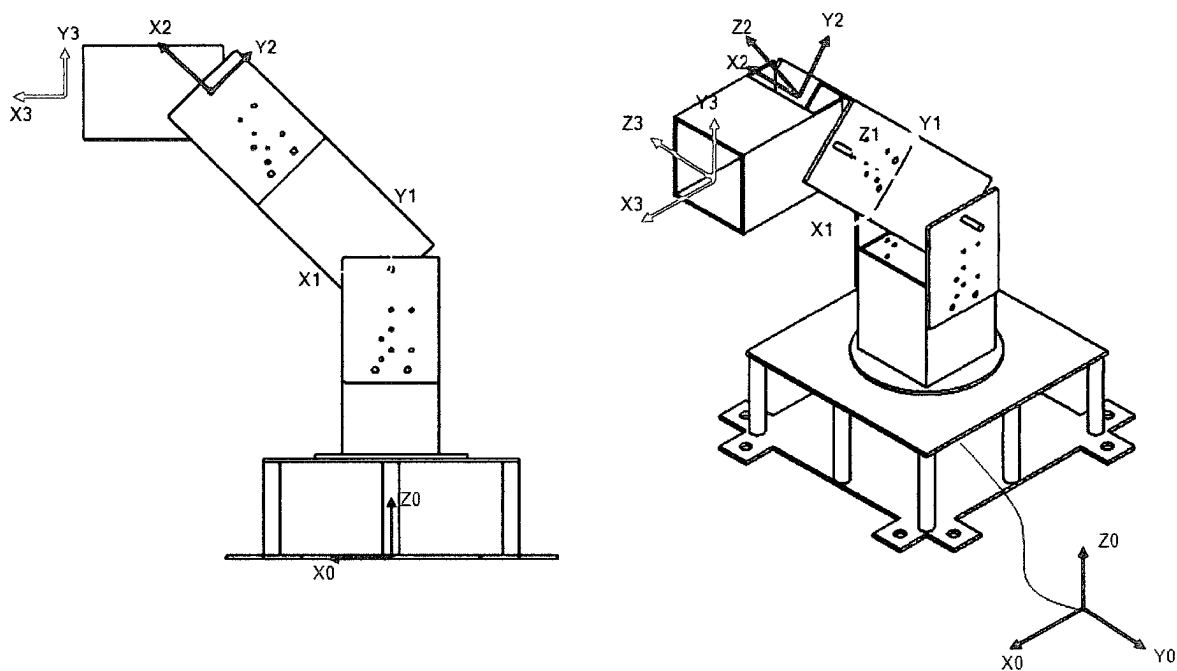


Figura 69 Sistemas coordenados de las articulaciones según Denavit-Hartenberg

WWW.BDIGITAL.ULA.VE

Una vez obtenidos los sistemas coordenados y siguiendo el algoritmo Denavit-Hartenberg, se calculan los parámetros D-H para el manipulador robótico en estudio. Los parámetros se muestran en la Tabla 10.

Tabla 10 Parámetros Denavit-Hartenberg para el manipulador robótico en estudio

Articulación / Parámetro	θ_i	d_i	a_i	α_i
Articulación 1	θ_1	230	0	90
Articulación 2	θ_2	0	200	0
Articulación 3	θ_3	0	130	0

4.2.1 Cinemática Directa

La cinemática directa, permite ubicar las posición del extremo del manipulador a partir de los valores articulares. Para ello, una vez se tienen los parámetros Denavit-Hartenberg se calculan

Licencia Creative Commons:

Atribución - No Comercial - Compartir Igual 3.0 Venezuela
(CC BY-NC-SA 3.0 VE)

las matrices de transformación homogénea para cada una de las articulaciones; para la articulación uno 0A_1 , ver ecuación (36); para la articulación dos 1A_2 , ver ecuación (37) y para la articulación tres 2A_3 , ver ecuación (38).

$${}^0A_1 = \begin{bmatrix} \cos \theta_1 & 0 & \sin \theta_1 & 0 \\ \sin \theta_1 & 0 & -\cos \theta_1 & 0 \\ 0 & 1 & 0 & 230 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (36)$$

$${}^1A_2 = \begin{bmatrix} \cos \theta_2 & -\sin \theta_2 & 0 & 200 \cos \theta_2 \\ \sin \theta_2 & \cos \theta_2 & 0 & 200 \sin \theta_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (37)$$

$${}^2A_3 = \begin{bmatrix} \cos \theta_3 & -\sin \theta_3 & 0 & 130 \cos \theta_3 \\ \sin \theta_3 & \cos \theta_3 & 0 & 130 \sin \theta_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (38)$$

Una vez obtenidas las matrices de transformación homogéneas de las articulaciones, se calcula la matriz de transformación homogénea para el manipulador completo 0A_3 , ver ecuación (39); con la cual se puede obtener la posición del extremo del manipulador robótico en el sistema de referencia de la base, partiendo de los valores de las articulaciones. Para la ecuación (39) se han sustituido las funciones trigonométricas seno por la letra “S”, y coseno por la letra “C” debido al reducido espacio que se tiene para expresar la misma. Para el cálculo de las ecuaciones que se muestran de aquí en adelante se realiza un programa en la herramienta computacional Maple®, donde se realizan todas las operaciones; el programa junto con las soluciones se puede ver en el Anexo A.

$${}^0A_3 = T = \begin{bmatrix} C\theta_1 C\theta_2 C\theta_3 - C\theta_1 S\theta_2 S\theta_3 & -C\theta_1 C\theta_2 S\theta_3 - C\theta_1 S\theta_2 C\theta_3 & S\theta_1 \\ S\theta_1 C\theta_2 C\theta_3 - S\theta_1 S\theta_2 S\theta_3 & -S\theta_1 C\theta_2 S\theta_3 - S\theta_1 S\theta_2 C\theta_3 & -C\theta_1 \\ S\theta_2 C\theta_3 + C\theta_2 S\theta_3 & -S\theta_2 S\theta_3 + C\theta_2 C\theta_3 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad (39)$$

$$\begin{bmatrix} 130C\theta_1 C\theta_2 C\theta_3 - 130C\theta_1 S\theta_2 S\theta_3 + 200C\theta_1 C\theta_2 \\ 130S\theta_1 C\theta_2 C\theta_3 - 130S\theta_1 S\theta_2 S\theta_3 + 200S\theta_1 C\theta_2 \\ \dots 130S\theta_2 C\theta_3 + 130C\theta_2 S\theta_3 + 230 + 200S\theta_2 \\ 1 \end{bmatrix}$$

De la ecuación anterior (39), los tres primeros términos de la última columna son los correspondientes a la posición del extremo del manipulador en el sistema coordenado de la base. Separando los términos para un mejor entendimiento se tiene que para cada una de las coordenadas X,Y,Z se tiene una ecuación en función de los valores articulares. Cada una de las tres ecuaciones se muestran a continuación.

$$X = 130C\theta_1 C\theta_2 C\theta_3 - 130C\theta_1 S\theta_2 S\theta_3 + 200C\theta_1 C\theta_2 \quad (40)$$

$$Y = 130S\theta_1 C\theta_2 C\theta_3 - 130S\theta_1 S\theta_2 S\theta_3 + 200S\theta_1 C\theta_2 \quad (41)$$

$$Z = 130S\theta_2 C\theta_3 + 130C\theta_2 S\theta_3 + 230 + 200S\theta_2 \quad (42)$$

4.2.2 Cinemática Inversa

La cinemática inversa, caso contrario a la cinemática directa, permite calcular los valores articulares a partir de una posición del extremo del manipulador robótico. Para este caso se trabaja con las matrices de transformación homogénea inversa de cada una de las articulaciones, que se muestran a continuación.

$${}^1A_0 = \begin{bmatrix} \cos \theta_1 & \sin \theta_1 & 0 & 0 \\ 0 & 0 & 1 & -230 \\ \sin \theta_1 & -\cos \theta_1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (43)$$

$${}^2A_1 = \begin{bmatrix} \cos \theta_2 & \sin \theta_2 & 0 & -200 \\ -\sin \theta_2 & \cos \theta_2 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (44)$$

$${}^3A_2 = \begin{bmatrix} \cos \theta_3 & \sin \theta_3 & 0 & -130 \\ -\sin \theta_3 & \cos \theta_3 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (45)$$

Una vez obtenidas las matrices de transformación homogénea inversa para cada una de las articulaciones, se manipulan de manera de conseguir las ecuaciones de los valores articulares en función de la posición del extremo del manipulador, para ello se parte de la ecuación (46).

$${}^1A_0 * T = {}^2A_1 * {}^3A_2 \quad (46)$$

Donde T viene dada por la ecuación (47), en donde la matriz 3x3 NOA es la matriz de orientación, el vector 3x1 P es el vector de posición, el vector 1x3 que es totalmente cero, es el vector de perspectiva, y finalmente el vector 1x1 que tiene por valor uno es el vector de escalado. Al resolver la ecuación (46) anterior se obtiene un sistema de doce ecuaciones con doce incógnitas, a partir de estas ecuaciones se puede obtener el valor de la articulación uno como se muestra en la ecuación (48).

$$T = \begin{bmatrix} N_x & O_x & A_x & P_x \\ N_y & O_y & A_y & P_y \\ N_z & O_z & A_z & P_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (47)$$

$$\theta_1 = \arctan\left(\frac{Py}{Px}\right) \quad (48)$$

Para conseguir los valores de las demás articulaciones se siguen manipulando las matrices de transformación homogénea inversa, como se muestra en la ecuación (49). De esta ecuación también se obtiene un sistema de doce ecuaciones con doce incógnitas del cual se consiguen los valores de las articulaciones dos y tres, ver ecuaciones (50) y (51) respectivamente.

$${}^2A_1 {}^1A_0 {}^0T = {}^3A_2 \quad (49)$$

$$\theta_2 = 2 * \arctan\left(\frac{\left(b \pm \sqrt{b^2 - c^2 + a^2}\right)}{a + c}\right)$$

Donde: (50)

$$a = \cos \theta_1 Px + \sin \theta_1 Py$$

$$b = Pz - 230$$

$$c = \frac{23100 + a^2 + b^2}{400}$$

$$\theta_3 = -\arctan\left(\frac{\sin \theta_2 \cos \theta_1 Px + \sin \theta_2 \sin \theta_1 Py - \cos \theta_2 Pz + 230 \cos \theta_2}{\cos \theta_1 \cos \theta_2 Px + \sin \theta_1 \cos \theta_2 Py + \sin \theta_2 Pz - 230 \sin \theta_2 - 200}\right) \quad (51)$$

Es de notar que para el valor articular θ_2 se pueden obtener dos valores, por consiguiente para el caso de θ_3 también se pueden obtener dos valores. Esto se debe a las configuraciones singulares

del brazo manipulador, donde un mismo punto del espacio se puede obtener de dos maneras, estas posiciones son conocidas con los nombres de codo arriba y codo abajo, mencionadas con anterioridad.

4.2.3 Matriz Jacobiana directa

La matriz Jacobiana directa permite conocer los valores de las velocidades de posicionamiento del manipulador robótico partiendo de conocer las velocidades articulares. En el caso del manipulador robótico en estudio, la matriz Jacobiana viene dada por la ecuación (52), en este caso se vuelve a sustituir las funciones trigonométricas seno por la letra “S” y coseno por la letra “C” debido a la extensión de la ecuación.

$$J_d = \begin{bmatrix} -130S\theta_1C\theta_2C\theta_3 + 130S\theta_1S\theta_2S\theta_3 - 200S\theta_1C\theta_2 & \dots \\ 130C\theta_1C\theta_2C\theta_3 - 130C\theta_1S\theta_2S\theta_3 + 200C\theta_1C\theta_2 & \dots \\ 0 & \dots \\ -130C\theta_1C\theta_2S\theta_3 - 130C\theta_1C\theta_2S\theta_3 - 200S\theta_2C\theta_1 & \dots \\ \dots -130S\theta_1S\theta_2C\theta_3 - 130S\theta_1C\theta_2S\theta_3 - 200S\theta_2S\theta_1 & \dots \\ 130C\theta_2C\theta_3 - 130S\theta_2S\theta_3 + 200C\theta_2 & \dots \\ -130C\theta_1C\theta_2S\theta_3 - 130C\theta_1S\theta_2C\theta_3 & \dots \\ \dots -130S\theta_1C\theta_2S\theta_3 - 130S\theta_1S\theta_2C\theta_3 & \dots \\ -130S\theta_2S\theta_3 + 130C\theta_2C\theta_3 & \dots \end{bmatrix} \quad (52)$$

4.2.4 Matriz Jacobiana inversa

La matriz Jacobiana inversa permite determinar las velocidades de las articulaciones basándose en la velocidad del posicionamiento del extremo del manipulador robótico. Se tienen dos maneras de obtener la matriz Jacobiana, la inversa directa que consiste en invertir directamente la ecuación de la Jacobiana directa, y la Jacobiana inversa de la cinemática inversa, donde se calcula la matriz Jacobiana a partir de la cinemática inversa. Para el primer caso, la matriz Jacobiana inversa directa está expresada por la ecuación (53). La matriz Jacobiana inversa de la cinemática inversa se muestra en el Anexo A debido a la extensión de la misma.

$$J_d = \begin{bmatrix} \frac{S\theta_1}{10(-20C\theta_2 - 13C\theta_2C\theta_3 - 13S\theta_2S\theta_3)} \\ \frac{-(S\theta_2S\theta_3 - C\theta_2C\theta_3)C\theta_1}{200S\theta_3} \dots \\ \frac{C\theta_1(-20C\theta_2 - 13C\theta_2C\theta_3 + 13S\theta_2S\theta_3)}{2600S\theta_3} \\ \frac{-C\theta_1}{10(-20C\theta_2 - 13C\theta_2C\theta_3 - 13S\theta_2S\theta_3)} \\ \dots \frac{-(S\theta_2S\theta_3 - C\theta_2C\theta_3)S\theta_1}{200S\theta_3} \dots \\ \frac{S\theta_1(-20C\theta_2 - 13C\theta_2C\theta_3 + 13S\theta_2S\theta_3)}{2600S\theta_3} \\ \dots \\ \dots \frac{S\theta_2C\theta_3 + C\theta_2S\theta_3}{S\theta_3} \\ \dots \frac{13S\theta_2C\theta_3 + 20S\theta_2 + 13C\theta_2S\theta_3}{-2600S\theta_3} \end{bmatrix} \tag{53}$$

4.2.5 Dinámica del manipulador propuesto

El cálculo de la dinámica del manipulador robótico propuesto se realiza mediante la formulación recursiva de Newton – Euler, por ser uno de los métodos más sencillos, a la vez que realizar los cálculos computacionalmente es de menor complejidad y más eficiente, ya que se utilizan operaciones vectoriales. Para desarrollar el cálculo de la dinámica del manipulador robótico se sigue el algoritmo desarrollado por Newton – Euler. (Barrientos, Peñín, Balaguer, & Aracil, Fundamentos de Robótica, 2007, págs. 232, 233, 234).

Se comienza por asignar los sistemas de referencia según el algoritmo de Denavit – Hartenberg, este procedimiento ya se realizó para el desarrollo de los cálculos anteriores. Luego se establecen las condiciones iniciales, ver ecuación (54).

$$\begin{aligned}
 {}^0\omega_0 &= [0 \ 0 \ 0]^T \\
 {}^0\dot{\omega}_0 &= [0 \ 0 \ 0]^T \\
 {}^0V_0 &= [0 \ 0 \ 0]^T \\
 {}^0\dot{V}_0 &= [0 \ 0 \ g_z]^T \\
 Z_0 &= [0 \ 0 \ 1]^T
 \end{aligned} \tag{54}$$

Donde:

${}^0\omega_0$ = Velocidad angular inicial

${}^0\dot{\omega}_0$ = Aceleración angular inicial

0V_0 = Velocidad lineal inicial

${}^0\dot{V}_0$ = Aceleración lineal inicial

Z_0 = Vector unitario de la gravedad

Luego se obtienen las coordenadas de cada eslabón con respecto al eslabón anterior.

$$\begin{aligned}
 {}^1P_1 &= \begin{bmatrix} 0 \\ 230 \sin \theta_1 \\ 230 \cos \theta_1 \end{bmatrix} \\
 {}^2P_2 &= \begin{bmatrix} 200 \\ 0 \\ 0 \end{bmatrix} \\
 {}^3P_3 &= \begin{bmatrix} 130 \\ 0 \\ 0 \end{bmatrix}
 \end{aligned} \tag{55}$$

Una vez obtenidas las coordenadas de cada eslabón con respecto al eslabón anterior, se ubican los centros de masa con respecto al sistema de referencia de cada eslabón. Para ello se utiliza el

prototipo virtual de cada uno de los eslabones ensamblado completamente con todas las piezas, y cada una de las piezas con sus materiales respectivos; luego con ayuda del programa CAD utilizado se obtiene el centro de gravedad para cada uno de los eslabones. Ver Anexo M

$$\begin{aligned}
 {}^1S_1 &= \begin{bmatrix} 0 \\ -82 \\ 0 \end{bmatrix} \\
 {}^2S_2 &= \begin{bmatrix} -86,78 \\ 0 \\ 13 \end{bmatrix} \\
 {}^3S_3 &= \begin{bmatrix} -78,8 \\ 0 \\ 0 \end{bmatrix}
 \end{aligned} \tag{56}$$

Posteriormente con la ayuda del programa CAD se obtienen las matrices de inercia de cada eslabón con respecto a su centro de masa, expresado en el sistema coordenado de cada eslabón, Ver anexo M:

$$\begin{aligned}
 {}^1I_1 &= \begin{bmatrix} 2422,309 & 0 & 0 \\ 4,285 & 1587,979 & 0 \\ -0,002 & -0,002 & 1474,055 \end{bmatrix} \\
 {}^2I_2 &= \begin{bmatrix} 1059,248 & 0 & 0 \\ 1,678 & 2089,822 & 0 \\ -99,320 & 43,420 & 1594,025 \end{bmatrix} \\
 {}^3I_3 &= \begin{bmatrix} 282,163 & 0 & 0 \\ 0 & 284,792 & 0 \\ 0 & 0 & 262,959 \end{bmatrix}
 \end{aligned} \tag{57}$$

El paso siguiente en el algoritmo de Newton – Euler es conseguir las matrices de rotación para los eslabones y sus inversas.

$${}^0A_1 = \begin{bmatrix} \cos \theta_1 & 0 & \sin \theta_1 \\ \sin \theta_1 & 0 & -\cos \theta_1 \\ 0 & 1 & 0 \end{bmatrix} \quad (58)$$

$${}^1A_2 = \begin{bmatrix} \cos \theta_2 & -\sin \theta_2 & 0 \\ \sin \theta_2 & \cos \theta_2 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (59)$$

$${}^2A_3 = \begin{bmatrix} \cos \theta_3 & -\sin \theta_3 & 0 \\ \sin \theta_3 & \cos \theta_3 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (60)$$

$${}^1A_0 = \begin{bmatrix} \cos \theta_1 & \sin \theta_1 & 0 \\ 0 & 0 & 1 \\ \sin \theta_1 & -\cos \theta_1 & 0 \end{bmatrix} \quad (61)$$

$${}^2A_1 = \begin{bmatrix} \cos \theta_2 & \sin \theta_2 & 0 \\ -\sin \theta_2 & \cos \theta_2 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (62)$$

$${}^3A_2 = \begin{bmatrix} \cos \theta_3 & \sin \theta_3 & 0 \\ -\sin \theta_3 & \cos \theta_3 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (63)$$

Se obtienen las velocidades angulares de los sistemas de referencia de los eslabones número uno, número dos y número tres.

$${}^1\omega_1 = \begin{bmatrix} 0 \\ \dot{\theta}_1 \\ 0 \end{bmatrix} \quad (64)$$

$${}^2\omega_2 = \begin{bmatrix} \sin \theta_2 \dot{\theta}_1 \\ \cos \theta_2 \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix} \quad (65)$$

$${}^3\omega_3 = \begin{bmatrix} \cos \theta_3 \sin \theta_2 \dot{\theta}_1 + \sin \theta_3 \cos \theta_2 \dot{\theta}_1 \\ -\sin \theta_3 \sin \theta_2 \dot{\theta}_1 + \cos \theta_3 \sin \theta_2 \dot{\theta}_1 \\ \dot{\theta}_2 + \dot{\theta}_3 \end{bmatrix} \quad (66)$$

Posteriormente se obtienen las aceleraciones angulares para cada uno de los sistemas de referencia de cada uno de los eslabones.

WWW.BDIGITAL.ULA.VE

$${}^1\dot{\omega}_1 = \begin{bmatrix} 0 \\ \ddot{\theta}_1 \\ 0 \end{bmatrix} \quad (67)$$

$${}^2\dot{\omega}_2 = \begin{bmatrix} \sin \theta_2 \ddot{\theta}_1 + \dot{\theta}_1 \dot{\theta}_2 \\ \cos \theta_2 \ddot{\theta}_1 \\ \ddot{\theta}_2 \end{bmatrix} \quad (68)$$

$${}^3\dot{\omega}_3 = \begin{bmatrix} \cos \theta_3 (\sin \theta_2 \ddot{\theta}_1 + \dot{\theta}_1 \dot{\theta}_2) + \sin \theta_3 \cos \theta_2 \ddot{\theta}_1 + \cos \theta_2 \dot{\theta}_1 \dot{\theta}_3 \\ -\sin \theta_3 (\sin \theta_2 \ddot{\theta}_1 + \dot{\theta}_1 \dot{\theta}_2) + \cos \theta_3 \cos \theta_2 \ddot{\theta}_1 - \sin \theta_2 \dot{\theta}_2 \dot{\theta}_3 \\ \ddot{\theta}_2 + \ddot{\theta}_3 \end{bmatrix} \quad (69)$$

Junto con las aceleraciones angulares se obtienen las aceleraciones lineales para cada uno de los sistemas de referencia de los eslabones.

$${}^1\dot{V}_1 = \begin{bmatrix} 0 \\ g_z \\ 0 \end{bmatrix} \quad (70)$$

$${}^2\dot{V}_2 = \begin{bmatrix} -200 \cos(\theta_2)^2 (\dot{\theta})^2 - 200 (\dot{\theta}_2)^2 + \sin \theta_2 g_z \\ 200 \ddot{\theta} + 200 \sin \theta_2 (\dot{\theta}_1)^2 \cos \theta_2 + \cos \theta_2 g_z \\ -200 \cos \theta_2 \ddot{\theta}_1 + 200 \sin \theta_2 \dot{\theta}_1 \dot{\theta}_2 \end{bmatrix} \quad (71)$$

La aceleración lineal del eslabón número tres se muestra en el Anexo B debido a la longitud de la misma. Después de obtenidas las aceleraciones lineales de los sistemas coordenados de los eslabones, se obtienen las aceleraciones lineales de los centros de gravedad de cada uno de los eslabones.

$${}^1a_1 = \begin{bmatrix} 2,807 (\dot{\theta}_1)^2 \\ g_z \\ 2,807 (\ddot{\theta}_1)^2 \end{bmatrix} \quad (72)$$

Las aceleraciones de los centros de gravedad del segundo y tercer eslabón se presentan en el Anexo B, por la longitud de las ecuaciones.

Posteriormente se calcula la fuerza ejercida sobre cada uno de los eslabones, para ello se determina primero la masa en cada uno de ellos, como se muestra en la Tabla 11. El peso se determina con todos los elementos que lo componen, es decir, motores, elementos de sujeción,

elementos de transmisión de potencia, etc. El cálculo de la fuerza ejercida sobre los eslabones se realiza de manera recursiva, y se comienza sobre el eslabón número dos ya que se asume que sobre eslabón número tres se tiene como condición inicial que no está actuando ninguna carga o fuerza. Las ecuaciones de la fuerza ejercida sobre los eslabones se muestran en el Anexo B.

Tabla 11 Peso de cada uno de los eslabones

Eslabón	Peso (Kg)
Eslabón 1	0,583
Eslabón 2	0,473
Eslabón 3	0,153

Finalmente, utilizando el mismo método de cálculo recursivo, se calcula el par ejercido sobre cada uno de los eslabones, así como también, el par ejercido sobre cada una de las articulaciones. Estas ecuaciones se muestran en el Anexo B debido a que la longitud de las mismas no permite mostrarlas en el presente formato.

WWW.BDIGITAL.ULA.VE

4.3 Comparación de cálculos

Para comprobar las ecuaciones obtenidas para la cinemática directa, se crea un rutina en el software matemático MATLAB ®, ver Anexo C; en la que se programan las ecuaciones obtenidas correspondientes a la cinemática directa del robot manipulador propuesto. Con la finalidad de poder comparar los resultados y comprobar que concuerdan, se crea otra rutina con la ayuda del “Robotic ToolBox for MATLAB ®” en la que se introducen los parámetros del robot manipulador propuesto, ver Anexo D. Para ambas rutinas se utilizan los mismos valores de entrada, para el caso de la cinemática directa se introducen valores de ángulos que toman las articulaciones y se obtiene una posición en el espacio del extremo del robot manipulador; luego se comparan los resultados obtenidos en la Tabla 12.

Tabla 12 Comparación de resultados para la cinemática directa

Valores articulares (rad) $(\theta_1, \theta_2, \theta_3)$	Propuesta propia (mm)	“Robotic ToolBox” (mm)
$(0 \ 0 \ 0)$	$\begin{bmatrix} 1 & 0 & 0 & 330 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 230 \\ 0 & 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & 0 & 330 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 230 \\ 0 & 0 & 0 & 1 \end{bmatrix}$
$(0 \ 1.5708 \ 0)$	$\begin{bmatrix} 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 1 & 0 & 0 & 560 \\ 0 & 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 1 & 0 & 0 & 560 \\ 0 & 0 & 0 & 1 \end{bmatrix}$
$(0 \ \pi/2 \ -\pi/2)$	$\begin{bmatrix} 1 & 0 & 0 & 130 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 430 \\ 0 & 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & 0 & 130 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 430 \\ 0 & 0 & 0 & 1 \end{bmatrix}$
$(0 \ \pi/2 \ \pi/2)$	$\begin{bmatrix} -1 & 0 & 0 & -130 \\ 0 & 0 & -1 & 0 \\ 0 & -1 & 0 & 430 \\ 0 & 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} -1 & 0 & 0 & -130 \\ 0 & 0 & -1 & 0 \\ 0 & -1 & 0 & 430 \\ 0 & 0 & 0 & 1 \end{bmatrix}$
$(\pi \ 0 \ 0)$	$\begin{bmatrix} -1 & 0 & 0 & -330 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 230 \\ 0 & 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} -1 & 0 & 0 & -330 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 230 \\ 0 & 0 & 0 & 1 \end{bmatrix}$
$(0.7854 \ -0.5231 \ 1.0910)$	$\begin{bmatrix} 0.5961 & -0.3803 & 0.7071 & 200.0042 \\ 0.5961 & -0.3803 & -0.7071 & 200.0049 \\ 0.5379 & 0.8430 & 0 & 200.0086 \\ 0 & 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & 0 & 200 \\ 0 & 1 & 0 & 200 \\ 0 & 0 & 1 & 200 \\ 0 & 0 & 0 & 1 \end{bmatrix}$
$(0.7854 \ -1.4181 \ 1.9656)$	$\begin{bmatrix} 0.6037 & -0.3681 & 0.7071 & 99.9978 \\ 0.6037 & -0.3681 & -0.7071 & 99.9981 \\ 0.5206 & 0.8538 & 0 & 99.9991 \\ 0 & 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & 0 & 100 \\ 0 & 1 & 0 & 100 \\ 0 & 0 & 1 & 100 \\ 0 & 0 & 0 & 1 \end{bmatrix}$

$(-8.3176 \quad -3.22 \quad 8.6734)$	$\begin{bmatrix} -0.3011 & -0.3306 & -0.8944 & 49.9989 \\ -0.6023 & -0.6612 & 0.4472 & 100.0050 \\ -0.7393 & 0.6734 & 0 & 149.9989 \\ 0 & 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & 0 & 50 \\ 0 & 1 & 0 & 100 \\ 0 & 0 & 1 & 150 \\ 0 & 0 & 0 & 1 \end{bmatrix}$
$(0.7854 \quad 0.9334 \quad -1.7058)$	$\begin{bmatrix} 0.5065 & 0.4935 & 0.7071 & 149.9997 \\ 0.5065 & 0.4935 & -0.7071 & 150.0002 \\ -0.6979 & 0.7162 & 0 & 300.0083 \\ 0 & 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & 0 & 150 \\ 0 & 1 & 0 & 150 \\ 0 & 0 & 1 & 300 \\ 0 & 0 & 0 & 1 \end{bmatrix}$
$(0.7854 \quad 1.1445 \quad -1.2378)$	$\begin{bmatrix} 0.7040 & 0.0659 & 0.7071 & 150.0017 \\ 0.7040 & 0.0659 & -0.7071 & 150.0023 \\ -0.0932 & 0.9957 & 0 & 399.9893 \\ 0 & 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & 0 & 150 \\ 0 & 1 & 0 & 150 \\ 0 & 0 & 1 & 400 \\ 0 & 0 & 0 & 1 \end{bmatrix}$
$(\pi/3 \quad \pi/3 \quad \pi/3)$	$\begin{bmatrix} -0.25 & -0.43 & 0.87 & 17.50 \\ -0.43 & -0.75 & -0.50 & 30.31 \\ 0.87 & -0.50 & 0 & 515.79 \\ 0 & 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} -0.25 & -0.43 & 0.87 & 17.50 \\ -0.43 & -0.75 & -0.50 & 30.31 \\ 0.87 & -0.50 & 0 & 515.79 \\ 0 & 0 & 0 & 1 \end{bmatrix}$

Se puede observar que se consigue la misma posición espacial para ambos casos, aún y cuando la matriz de orientación no sea igual. Esto se debe a la manera como el “Robotics Toolbox for MATLAB®” realiza los cálculos, utiliza métodos recursivos para buscar una solución, y muestra la primera que consigue, sin tener oportunidad de poder seleccionar.

Para la comprobación de la cinemática inversa se crean tres rutinas en MATLAB ® con las ecuaciones de la cinemática inversa, una rutina donde se programan las ecuaciones de la cinemática inversa obtenidas a partir de las matrices homogéneas, y una rutina de las ecuaciones obtenidas por métodos geométricos. Finalmente se crea una rutina utilizando el “Robotic Toolbox for MATLAB®”. Para los tres casos se utilizan los mismos valores de entrada, en este caso se introduce una ubicación en el espacio por medio de coordenadas, y se obtienen los valores de las articulaciones. La comparación entre los tres métodos se puede observar en la Tabla 13.

Tabla 13 Comparación de las cinemáticas inversas

Ubicación (mm) (x,y,z)	C. Inversa 01	C. Inversa 02	“Robotics toolbox”
(330, 0, 230)	[0, 0, 0]	[0, 0, 0]	[0, 0, 0]
(0, 0, 560)	NA	[NA, 1.5708, 0]	[0, 1.5708, 0]
(-130, 0, 430)	[0, 1.5708, 1.5708]	[0, 1.5708, -1.5708]	[0, -3.5594, 4.7124]
(-330, 0, 230)	[0, NA, NA]	[0, 0, 0]	NA
(200, 200, 200)	[0.7854, -0.5232, 1.0911]	[0.7854, -0.5232, 1.0911]	[0.7854, -0.5231, 1.0910]
(100, 100, 100)	[0.7854, -0.0686, 1.1760]	[0.7854, -0.2958, -1.1760]	[0.7854, -1.4181, 1.9656]
(50, 100, 150)	[1.1071, -1.3228, -0.7513]	[1.1071, -0.3289, -0.7513]	[4.2487, -3.2220, 2.3901]
(150, 150, 300)	[0.7854, -0.2959, -1.4358]	[0.7854, 0.8534, -1.4358]	[0.7854, 0.9334, -1.7058]
(150, 150, 400)	[0.7854, 1.1445, -1.2378]	[0.7854, 1.1445, -1.2378]	[0.7854, 1.1445, -1.2378]
(17.5, 30.31, 515.79)	[1.0472, 1.8507, -1.0472]	[1.0472, 1.8507, -1.0472]	[1.0472, 1.8506, -1.0471]

Se puede observar que la mayoría de los resultados obtenidos coinciden, sin embargo existen casos puntuales en los cálculos hechos con las ecuaciones obtenidas, donde se consigue una división por cero (representados en la tabla con la palabra NA), razón por la que no se llega a un resultado, pudiéndose deber esto que se encuentra una posición singular. Existen otros casos donde los resultados no coinciden, esto se debe a la configuración que adopta el manipulador, teniendo en cuenta que en muchos casos se puede obtener una misma ubicación espacial con diferentes valores articulares del robot manipulador.

Para la comprobación del cálculo de la Jacobiana directa, se crea una rutina en MATLAB ® donde se programa la Jacobiana directa obtenida anteriormente. Ver Anexo E. Para comparar estos cálculos se utiliza la función para la Jacobiana directa del “Robotics Tollbox for MATLAB®”. Como datos de entrada se utiliza en cada caso valores articulares, y se obtienen las matrices Jacobianas directas. En la Tabla 14 se muestra la comparación entre los dos métodos utilizados.

Tabla 14 Comparación de las matrices Jacobianas directas

Valores articulares (rad) (q1 q2 q3)	Jacobiana directa calculada	Jacobiana directa "Robotics Toolbox"
(0 0 0)	$\begin{bmatrix} 0 & 0 & 0 \\ 330 & 0 & 0 \\ 0 & 330 & 130 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 0 \\ 330 & 0 & 0 \\ 0 & 330 & 130 \end{bmatrix}$
(0 1.5708 0)	$\begin{bmatrix} 0 & -330 & -130 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & -330 & -130 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$
(0 $\pi/2$ $-\pi/2$)	$\begin{bmatrix} 0 & -200 & 0 \\ 130 & 0 & 0 \\ 0 & 130 & 130 \end{bmatrix}$	$\begin{bmatrix} 0 & -200 & 0 \\ 130 & 0 & 0 \\ 0 & 130 & 130 \end{bmatrix}$
(0 $\pi/2$ $\pi/2$)	$\begin{bmatrix} 0 & -200 & 0 \\ -130 & 0 & 0 \\ 0 & -130 & -130 \end{bmatrix}$	$\begin{bmatrix} 0 & -200 & 0 \\ -130 & 0 & 0 \\ 0 & -130 & -130 \end{bmatrix}$
(π 0 0)	$\begin{bmatrix} 0 & 0 & 0 \\ -330 & 0 & 0 \\ 0 & 330 & 130 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 0 \\ -330 & 0 & 0 \\ 0 & 330 & 130 \end{bmatrix}$
(0.7854 -0.5231 1.0910)	$\begin{bmatrix} -200 & 21.2071 & -49.4424 \\ 200.0042 & 21.2071 & -49.4424 \\ 0 & 282.8491 & 109.5942 \end{bmatrix}$	$\begin{bmatrix} -200 & 21.2071 & -49.4424 \\ 200.0042 & 21.2071 & -49.4424 \\ 0 & 282.8491 & 109.5942 \end{bmatrix}$
(0.7854 -1.4181 1.9656)	$\begin{bmatrix} -99.9981 & 91.9243 & -47.8513 \\ 99.9981 & 91.9243 & -47.8513 \\ 0 & 141.4184 & 110.9977 \end{bmatrix}$	$\begin{bmatrix} -99.9981 & 91.9243 & -47.8513 \\ 99.9981 & 91.9243 & -47.8513 \\ 0 & 141.4184 & 110.9977 \end{bmatrix}$
(-8.3176 -3.22 8.6734)	$\begin{bmatrix} -100.0050 & -35.7755 & -42.9770 \\ 49.9989 & -71.5562 & -85.9602 \\ 0 & -111.8074 & 87.5432 \end{bmatrix}$	$\begin{bmatrix} -100.0050 & -35.7755 & -42.9770 \\ 49.9989 & -71.5562 & -85.9602 \\ 0 & -111.8074 & 87.5432 \end{bmatrix}$
(0.7854 0.9334 -1.7058)	$\begin{bmatrix} -150.0002 & -49.5032 & 64.1495 \\ 149.9997 & -49.5032 & 64.1495 \\ 0 & 212.1319 & 93.1109 \end{bmatrix}$	$\begin{bmatrix} -150.0002 & -49.5032 & 64.1495 \\ 149.9997 & -49.5032 & 64.1495 \\ 0 & 212.1319 & 93.1109 \end{bmatrix}$

$(0.7854 \quad 1.1445 \quad -1.2378)$	$\begin{bmatrix} -150.0023 & -120.2004 & 8.5640 \\ 150.0017 & -120.2008 & 8.5641 \\ 0 & 212.1349 & 129.4346 \end{bmatrix}$	$\begin{bmatrix} -150.0023 & -120.2004 & 8.5640 \\ 150.0017 & -120.2008 & 8.5641 \\ 0 & 212.1349 & 129.4346 \end{bmatrix}$
$(\pi/3 \quad \pi/3 \quad \pi/3)$	$\begin{bmatrix} -30.3101 & -142.8935 & -56.2913 \\ 17.4994 & -247.5003 & -97.4999 \\ 0 & 34.9990 & -65.006 \end{bmatrix}$	$\begin{bmatrix} -30.3109 & -142.8942 & -56.2917 \\ 17.5000 & -247.5000 & -97.5000 \\ 0 & 35.000 & -65.000 \end{bmatrix}$

Como se observa en la tabla anterior, que tanto los resultados de la matriz Jacobiana directa obtenida con las ecuaciones calculadas, como los resultados obtenidos a través del “Robotic toolbox for MATLAB ®” coinciden para todas las posiciones.

Luego de obtenidos los resultados de la matriz Jacobiana directa, se obtienen resultados para la matriz Jacobiana inversa. Para ello se realizan tres rutinas en MATLAB ®, cada una de estas rutinas está programada con un método distinto, según la obtención de la matriz Jacobiana inversa. El primer método es invirtiendo la matriz Jacobiana directa (Jacobiana inversa 01), el segundo método es conseguir la matriz Jacobiana inversa a partir de las matrices homogéneas inversas (Jacobiana inversa 02). Y el tercer método es a partir de la cinemática inversa conseguida geoméricamente (Jacobiana inversa 03), ver Anexo F, Anexo G, Anexo H respectivamente. En la Tabla 15 se muestran los resultados.

Tabla 15 Comparación resultados Jacobianas inversas

Posición (X,Y,Z)	Jacobiana Inversa 01	Jacobiana Inversa 02	Jacobiana Inversa 03
$(200; 200; 200)$	$\begin{bmatrix} -0.0025 & 0.0025 & 0 \\ -0.0028 & -0.0028 & 0.0040 \\ 0.0087 & 0.0087 & -0.0013 \end{bmatrix}$	$\begin{bmatrix} -0.0025 & 0.0025 & 0 \\ -0.0028 & -0.0028 & 0.0040 \\ 0.0087 & 0.0087 & -0.0013 \end{bmatrix}$	$\begin{bmatrix} -0.0025 & 0.0025 & 0 \\ -0.0028 & -0.0028 & 0.0040 \\ 0.0087 & 0.0087 & -0.0013 \end{bmatrix}$
$(100; 100; 100)$	$\begin{bmatrix} -0.0027 & 0.0027 & 0 \\ 0.0017 & 0.0017 & 0.0048 \\ -0.0076 & -0.0076 & -0.0043 \end{bmatrix}$	$\begin{bmatrix} -0.0050 & 0.0050 & 0 \\ 0.0017 & 0.0017 & 0.0048 \\ 0.0042 & 0.0042 & -0.0054 \end{bmatrix}$	$\begin{bmatrix} -0.0050 & 0.0050 & 0 \\ 0.0010 & 0.0010 & 0.0057 \\ 0.0042 & 0.0042 & -0.0054 \end{bmatrix}$

(50;100;150)	$\begin{bmatrix} -0.0031 & 0.0016 & 0 \\ 0.0022 & 0.0044 & 0.0054 \\ -0.0072 & -0.0145 & -0.0063 \end{bmatrix}$	$\begin{bmatrix} -0.0080 & 0.0040 & 0 \\ 0.0022 & 0.0044 & 0.0054 \\ 0.0028 & 0.0056 & -0.0045 \end{bmatrix}$	$\begin{bmatrix} -0.0080 & 0.0040 & 0 \\ 0.0008 & 0.0017 & 0.0076 \\ 0.0028 & 0.0056 & -0.0045 \end{bmatrix}$
(17.5; 30.3109; 515.7884)	$\begin{bmatrix} -0.0247 & 0.0143 & 0 \\ -0.0020 & -0.0035 & -0.0042 \\ 0.0008 & 0.0013 & 0.0127 \end{bmatrix}$	$\begin{bmatrix} -0.0247 & 0.0143 & 0 \\ -0.0020 & -0.0035 & -0.0042 \\ 0.0008 & 0.0013 & 0.0127 \end{bmatrix}$	$\begin{bmatrix} -0.0247 & 0.0143 & 0 \\ -0.0020 & -0.0035 & -0.0042 \\ 0.0008 & 0.0013 & 0.0127 \end{bmatrix}$

De la tabla anterior se observa que solamente en dos oportunidades, la primera y la última, los tres métodos para la matriz Jacobiana coinciden totalmente; para el resto de los casos sólo coinciden los resultados de las matrices Jacobianas inversas calculadas a partir de las inversas de las matrices homogéneas, y la que se calcula a partir de la cinemática inversa geométrica. Esta diferencia se debe al cálculo de la inversión de la matriz Jacobiana directa.

4.3.1 Trayectorias

Una trayectoria es el recorrido o camino que sigue el robot manipulador para trasladarse desde una posición inicial hasta una posición final. Son varios los caminos que se pueden seguir para ir de una posición a otra, sin embargo, son los recorridos más sencillos los que se suelen implementar en el uso de robots comerciales. Las trayectorias pueden ser punto a punto, donde se divide todo el recorrido en varios trayectos más cortos; y las trayectorias continuas que son manejadas por ecuaciones matemáticas.

Con las ecuaciones obtenidas para la cinemática inversa, se plantean tres trayectorias para observar el comportamiento de los valores articulares. Las trayectorias planteadas son trayectorias punto a punto, que según Barrientos (2007) son el tipo de trayectoria más utilizado para la programación de los manipuladores robóticos, por su sencillez en la implementación, programación y aplicación. Para llevar a cabo este procedimiento se crean dos programas en MATLAB®, uno para la cinemática inversa obtenida a partir de las matrices homogéneas de

transformación, y un programa para la cinemática inversa obtenida geoméricamente. Ver Anexos I y Anexo J. Ambos programas muestran los resultados para tres trayectorias distintas.

La primera trayectoria es una línea recta sobre el eje X, y con una altura constante (sobre el eje Z = 130 mm), es decir, solamente cambia el valor de la coordenada X, que comienza en X = 100 mm, y termina en X = 200 mm, con pasos intermedios cada 10 mm. La Figura 70 y la Figura 71 muestran los resultados de los valores articulares para la trayectoria presentada. En las gráficas se representan los dos resultados obtenidos para cada ángulo, y se denotan como theta2a, theta2b, theta3a y theta3b.

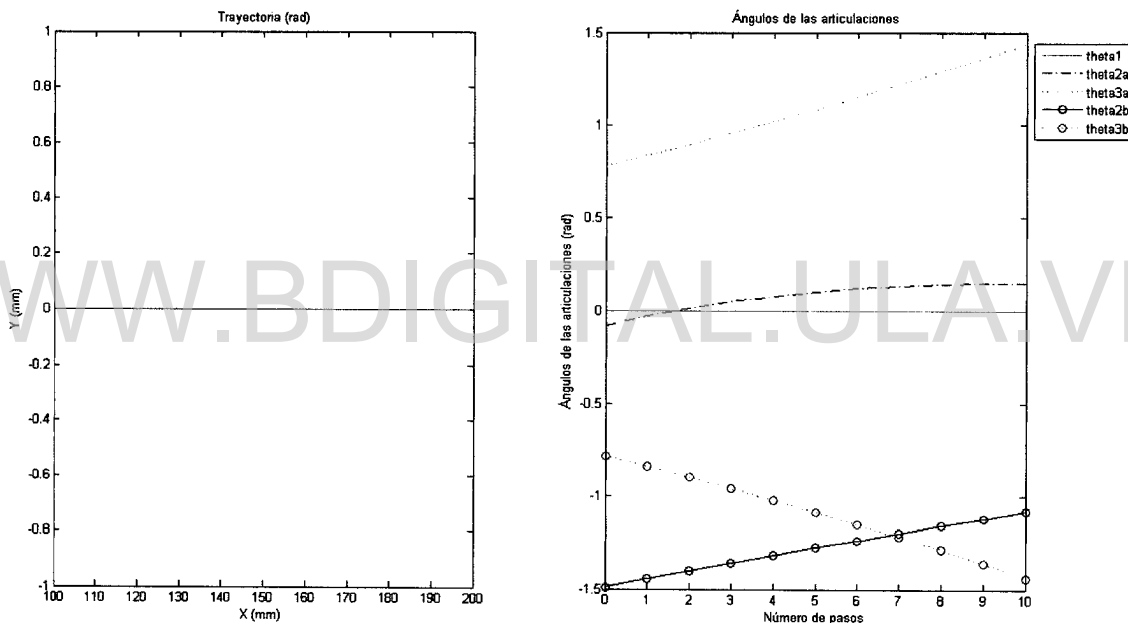


Figura 70 Resultados de la trayectoria 01 para la cinemática a partir de las matrices homogéneas.

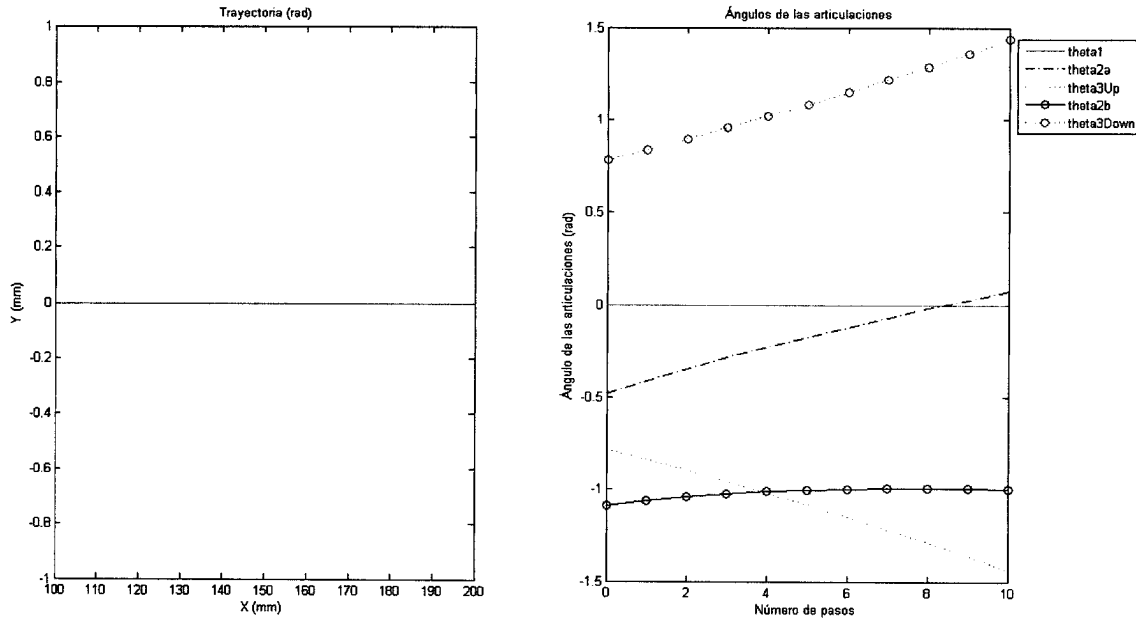


Figura 71 Resultados de la trayectoria 01 para la cinemática calculada geoméricamente.

En ambos casos, a pesar de diferir un poco en los valores articulares, se observa que se consigue el mismo resultado, seguir la trayectoria, y lo realizan con movimientos continuos sin cambios bruscos en los valores articulares. Adicionalmente se puede observar que los valores para la articulación número uno se mantienen constantes con valor de cero, como era de esperarse por la forma de la trayectoria. En las gráficas se muestran cinco valores articulares, esto se debe a las dos posibles soluciones que se pueden obtener para los valores de las articulaciones dos y tres.

La segunda trayectoria realizada, es una línea paralela al eje Y, donde se mantienen constantes los valores para las coordenadas X=50 mm y Z=130 mm. Las coordenadas Y, toman valor desde Y=0 hasta Y=200, con valores intermedios cada 20 mm. La Figura 72 y la Figura 73 muestran los resultados para la segunda trayectoria propuesta.

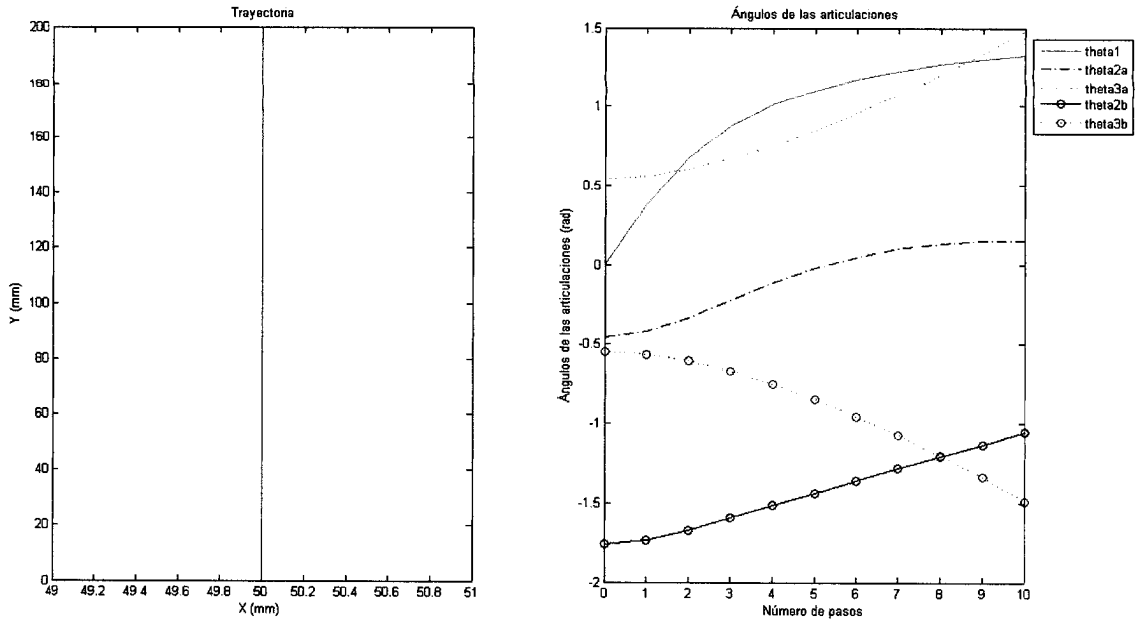


Figura 72 Resultados de la trayectoria 02 para la cinemática a partir de las matrices homogéneas.

WWW.BDIGITAL.ULA.VE

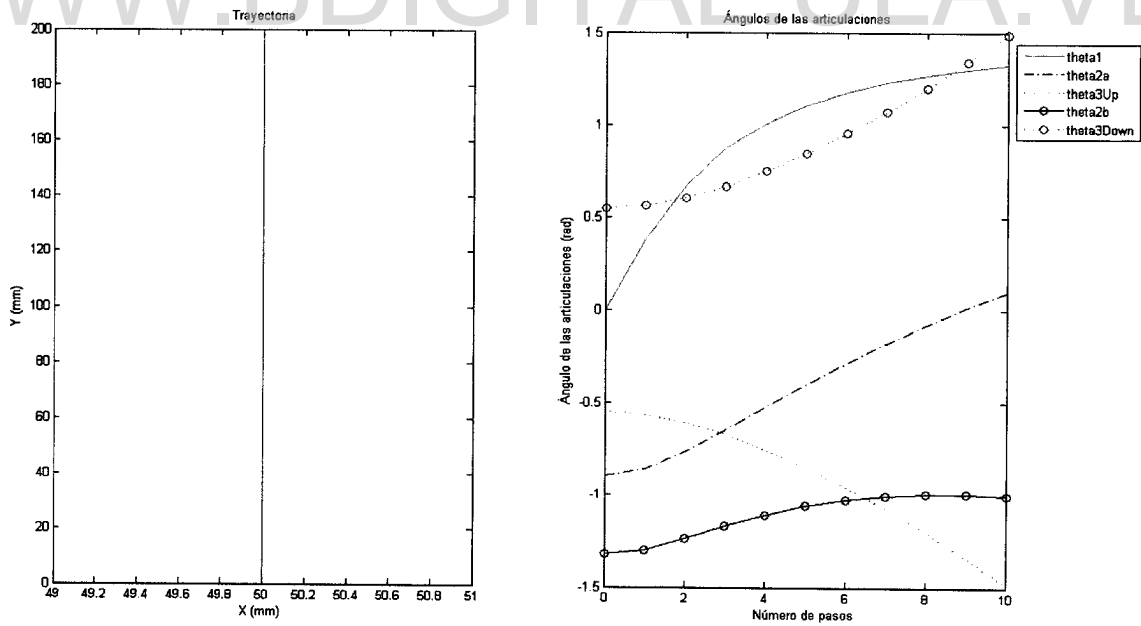


Figura 73 Resultados de la trayectoria 02 para la cinemática calculada geoméricamente.

Para el caso de la segunda trayectoria se puede observar como varían los valores articulares de manera continua y sin cambios bruscos, lo que se traduce en un movimiento constante por parte del robot manipulador.

Se presenta una tercera trayectoria, esta vez es una línea recta en el espacio que viene definida por $\frac{150-x}{1} = \frac{150-y}{1} = \frac{50-z}{1}$, y se utilizan valores para la coordenada X desde X=100 mm hasta X=150 mm con pasos intermedios de 5 mm. La Figura 74 y la Figura 75 muestran los resultados para la tercera trayectoria.

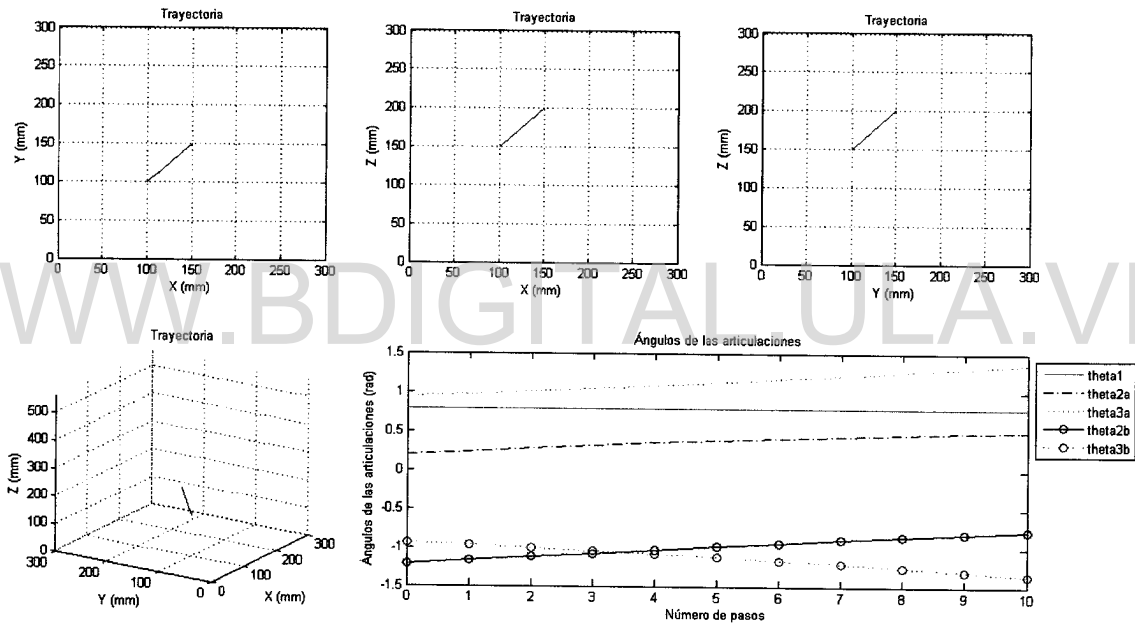


Figura 74 Resultados de la trayectoria 03 para la cinemática a partir de las matrices homogéneas.

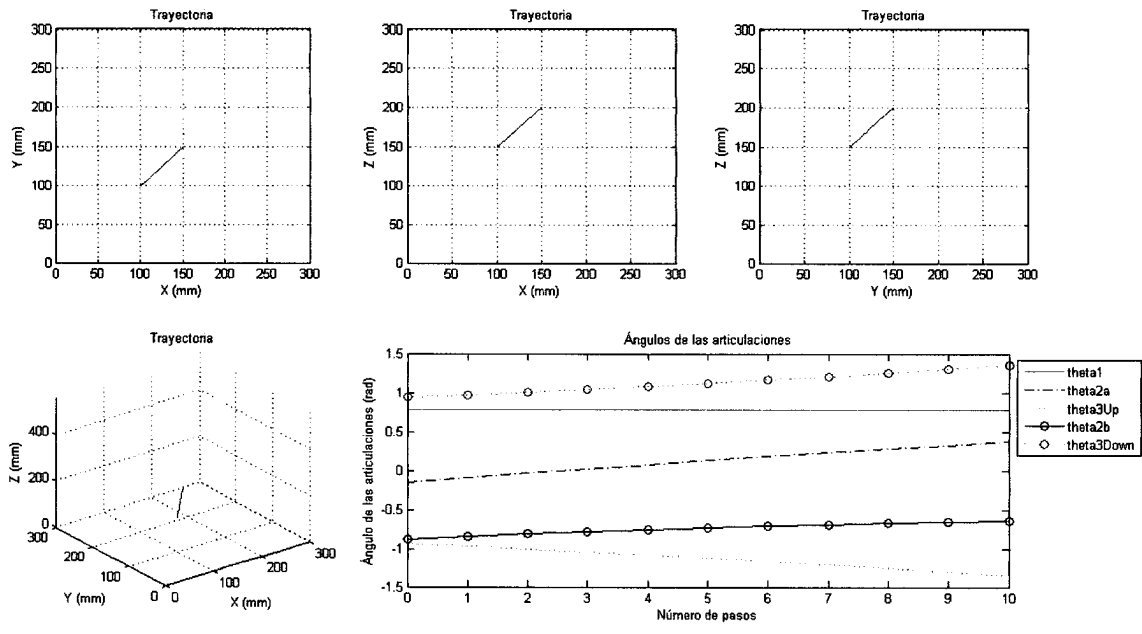


Figura 75 Resultados de la trayectoria 02 para la cinemática calculada geoméricamente.

De igual manera que en los casos anteriores, se puede observar como los valores articulares realizan una trayectoria suave y sin cambios bruscos de orientación.

Por último, se programan las tres trayectorias utilizadas anteriormente, haciendo uso del “Robotics Toolbox for MATLAB” y de esta manera observar el comportamiento de los valores articulares para el robot manipulador propuesto. La Figura 76 muestra los resultados obtenidos con el programa realizado, el cual puede ser observado en el Anexo K.

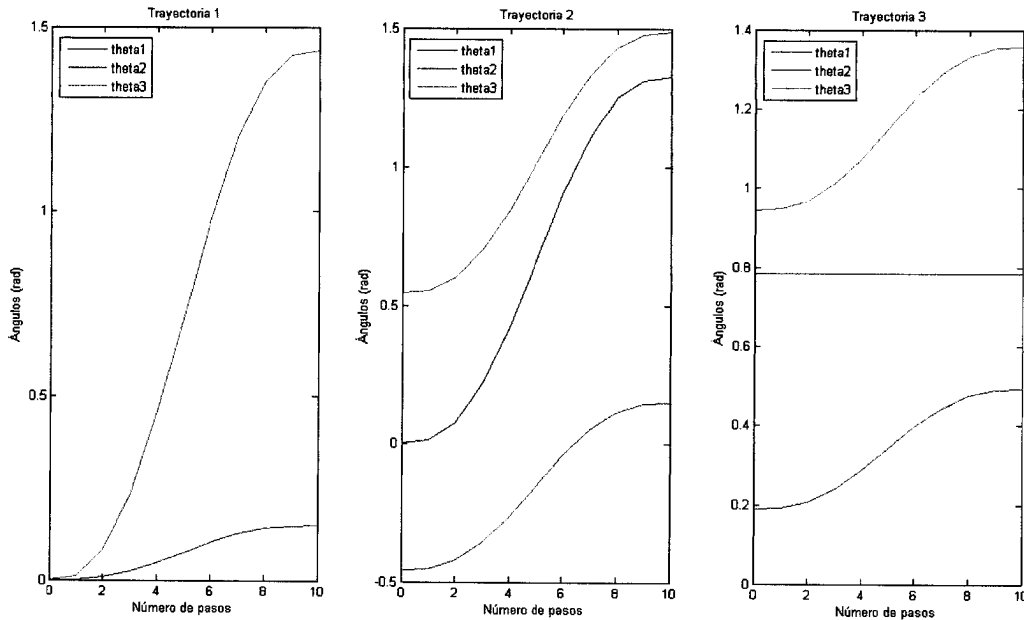


Figura 76 Resultados obtenidos con el “Robotic Toolbox for MATLAB”

Aún y cuando las gráficas muestran un seguimiento de trayectoria continuo, no es exactamente igual al calculado por las ecuaciones obtenidas para el robot manipulador propuesto. Estas desigualdades se deben a la forma del cálculo de la trayectoria hecha por el “Robotic Toolbox for MATLAB”, que utiliza un polinomio de grado 7 para el cálculo de la trayectoria, además de utilizar métodos de interpolación para calcular el recorrido entre cada uno de los puntos de utilizados para la trayectoria.

Las trayectorias mostradas anteriormente, realizadas por el robot manipulador propuesto, muestran cambios suaves y secuenciales de los valores articulares, esto no quiere decir que no se puedan presentar configuraciones singulares del robot, donde se presente cambios bruscos en los valores que toman los ángulos de las articulaciones, para seguir una trayectoria.

4.3.2 Dinámica

En este apartado se muestran los cálculos realizados para la determinación de los torques requeridos para el movimiento del robot manipulador en estudio. Para ello se realiza un programa en MATLAB ® donde se incorporan las ecuaciones obtenidas por medio del algoritmo de Newton-Euler para el cálculo de la dinámica del robot manipulador, ver anexo L.

Se realiza una primera corrida para el programa, donde se utiliza la primera trayectoria como guía. En la Figura 77 se muestran los resultados obtenidos.

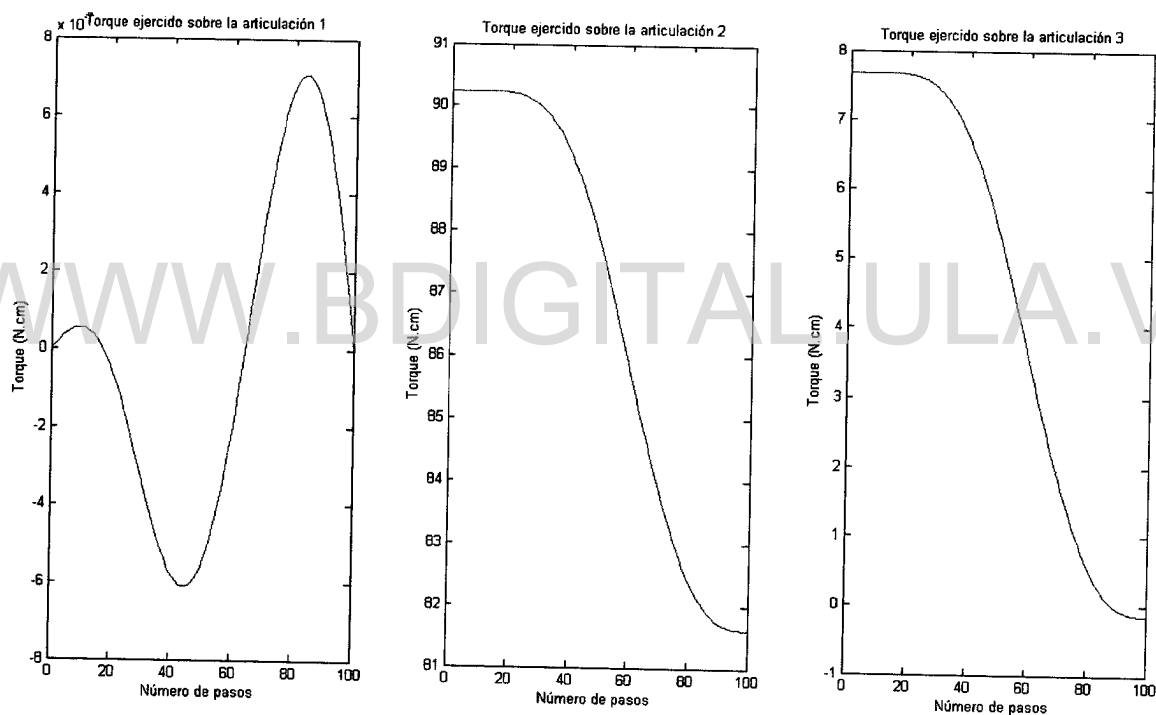


Figura 77 Torques obtenidos para la trayectoria 1.

Se realizan los cálculos nuevamente para la segunda trayectoria, obteniendo los resultados que se muestran en la Figura 78.

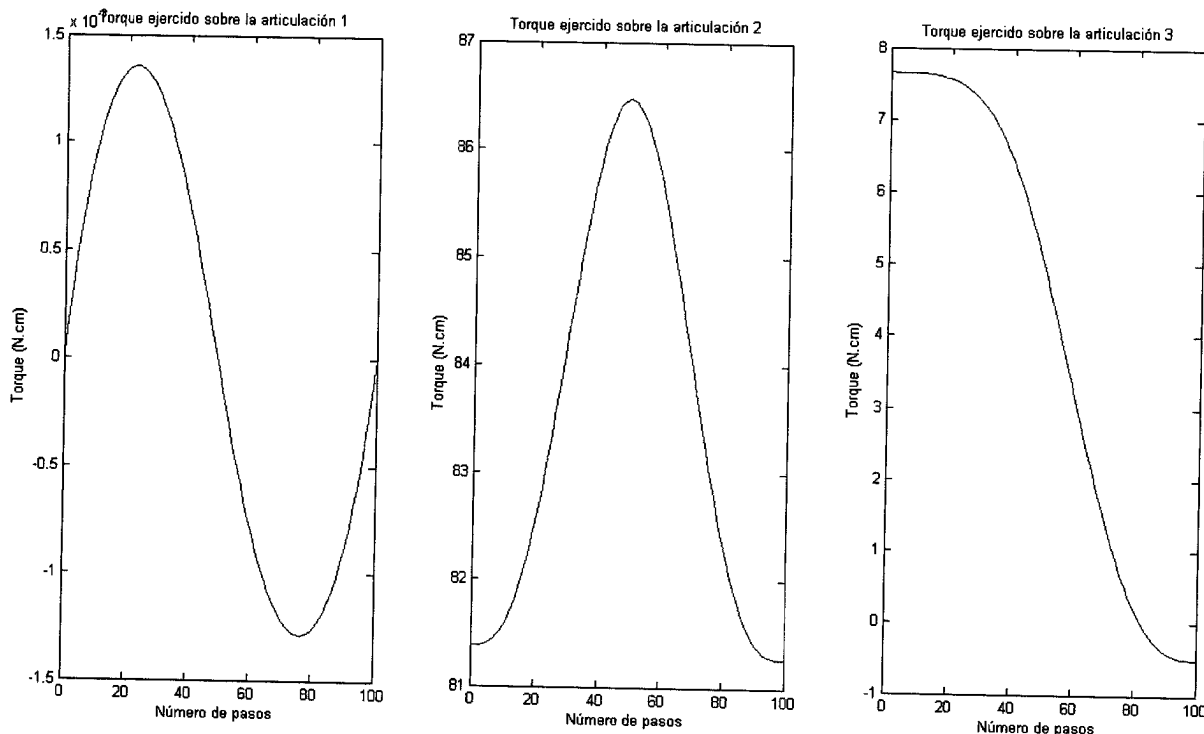


Figura 78 Torques obtenidos para la trayectoria 2.

Se realiza también el cálculo de los torques aplicados a las articulaciones para el caso de la tercera trayectoria. Los resultados se muestran en la Figura 79.

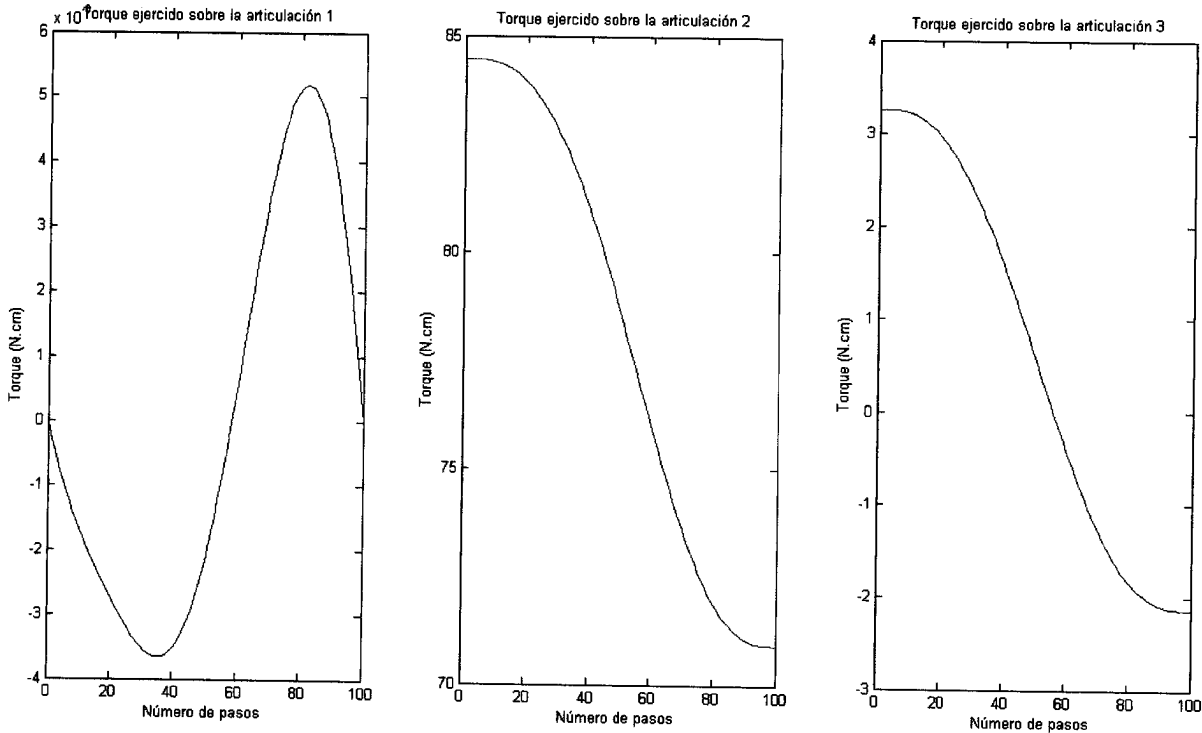


Figura 79 Torque obtenidos para la trayectoria 3.

Por último, se realiza un programa con el uso del “Robotics Toolbox for MATLAB” para el cálculo de los torques ejercidos sobre las articulaciones para las trayectorias mencionadas anteriormente, y se comparan con los resultados obtenidos anteriormente. El programa realizado se puede observar en el Anexo K. Los resultados obtenidos se pueden observar en las Figura 80, Figura 81 y Figura 82

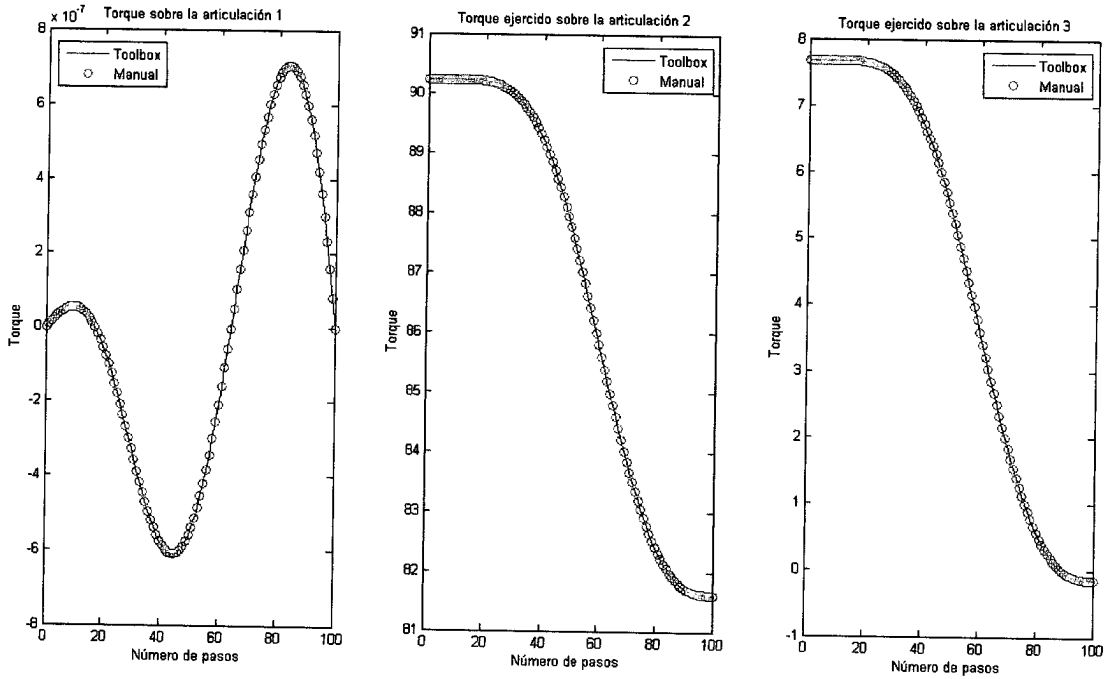


Figura 80 Comparación de los Torques obtenidos con el "Robotics toolbox for MATLAB" y las ecuaciones de Newton-Euler, para la trayectoria 1.

WWW.BDIGITAL.ULA.VE

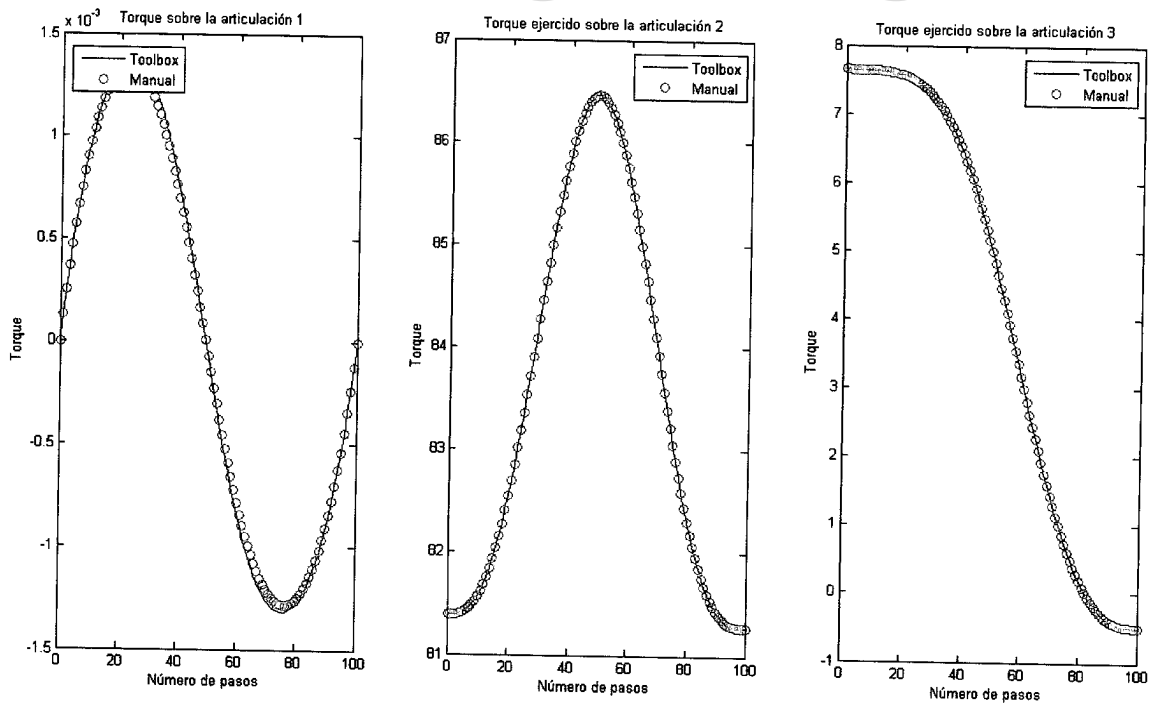


Figura 81 Comparación de los Torques obtenidos con el "Robotics toolbox for MATLAB" y las ecuaciones de Newton-Euler, para la trayectoria 2.

Licencia Creative Commons:
 Atribución - No Comercial - Compartir Igual 3.0 Venezuela
 (CC BY-NC-SA 3.0 VE)

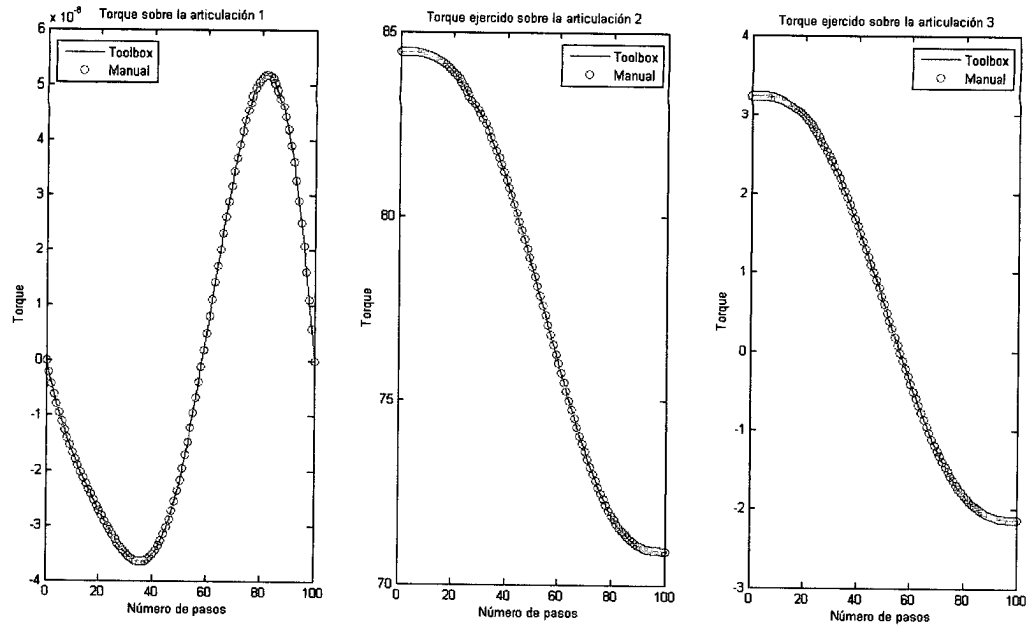


Figura 82 Comparación de los Torques obtenidos con el "Robotics toolbox for MATLAB" y las ecuaciones de Newton-Euler, para la trayectoria 3.

Se puede notar de la figura anterior que los torques aplicados no son constantes como en los casos anteriores, esto se debe a que el "Robotics Toolbox for MATLAB" no utiliza velocidades constantes para el seguimiento de la trayectoria; va aumentando la velocidad a medida que comienza el recorrido, y luego disminuye la velocidad cuando lo está terminando.

4.4 Herramienta computacional para la simulación y programación del manipulador robótico

Una vez realizados el diseño y fabricación del prototipo del manipulador robótico propuesto, es necesaria una herramienta computacional capaz de simular los movimientos y tareas que realizará el manipulador, así como también que tenga la opción de su programación. En tal sentido, se plantea el desarrollo de un trabajo que sirva como tesis de grado, donde se desarrolle la herramienta mencionada. Dicho trabajo, al ser derivado del presente estudio, también estará

enmarcado dentro de la línea de investigación adscrita al Consejo de Desarrollo Científico Humanístico y Tecnológico.

El proyecto es desarrollado en la escuela de Ingeniería de Sistemas de la Facultad de Ingeniería de la Universidad de Los Andes, y lleva por título “Desarrollo de una herramienta computacional para la simulación y programación de un manipulador robótico de tres grados de libertad”, sus autores son César Arismendi, Iñaki Aguirre y Judith Barrios.

El proyecto comienza con una revisión de las herramientas computacionales existentes, y que pueden ser de tipo comercial, de investigación, de fines didácticos, etc. La reseña muestra además el lenguaje de programación utilizado, el tipo de interfaz que tienen, si es de código abierto o cerrado. Luego de la revisión se realiza una tabla comparativa donde se muestran las ventajas y desventajas de cada una de las herramientas computacionales revisadas, ver Tabla 16.

Tabla 16 Tabla comparativa de simuladores.

Nombre del simulador	Ventajas	Desventajas
Robolab II	<ul style="list-style-type: none"> - Representación tridimensional - Controles cinemáticos completos - Permite la planificación de movimientos - Interactúa con robots 	<ul style="list-style-type: none"> - De código cerrado - Solo se puede utilizar en línea - Aplicable a robots específicos
Simrobot	<ul style="list-style-type: none"> - Representación tridimensional 	<ul style="list-style-type: none"> - Difícil de instalar - Solo se ejecuta en Windows 3.1.x, 95 y NT - No interactúa con robot real - No implementa modelos cinemáticos
Camelot	<ul style="list-style-type: none"> - Representación tridimensional - Permite cargar el modelo 	<ul style="list-style-type: none"> - Software comercial y de código cerrado

	tridimensional del robot - Permite la planificación de movimientos	- Solo para Windows - No permite interacción con el robot - Es costoso
EasyRob	- Representación tridimensional - Permite la planificación de movimientos - Permite cargar el modelo tridimensional del robot	- Software comercial y de código cerrado - Es costoso - Solo para Windows - La versión demostración parpadea en cada movimiento realizado
Robot3D	- Permite la planificación de movimientos - Permite Interactuar con robot - Permite calibrar los robots	- Software comercial y de código cerrado - Aplicable a robots específicos - Solo para Windows - Representación gráfica limitada
RoboSim	- Permite la planificación de movimientos	- Representación gráfica limitada - Parpadea al realizar un movimiento - Código cerrado - Solo se puede utilizar en línea
QuickTeach	- Representación tridimensional	- Software comercial y de código cerrado - Es costoso
ARM control	- Código abierto - Interactúa con el robot	- No implementa modelos cinemáticos - No presenta representación gráfica del robot
JRoboOP	- Representación tridimensional - Implementa el modelo cinemático y dinámico - Código abierto	- Aplicable a un robot específico - No permite interacción con el robot - El robot modelado es ficticio
Java 3D RoboSim	- Representación tridimensional - Código abierto	- No modela la cinemática del robot - Aplicable a un robot específico

Licencia Creative Commons:

Atribución - No Comercial - Compartir Igual 3.0 Venezuela
(CC BY-NC-SA 3.0 VE)

	- Permite la planificación de movimientos	- No implementa modelos cinemáticos
--	---	-------------------------------------

Fuente: (Arismendi, Aguirre, & Barrios, 2008, págs. 23, 24)

Luego de revisada la tabla anterior, y tomando en consideración aspectos como que la herramienta debiera ser multiplataforma, es decir que pueda correr en casi cualquier sistema operativo (Windows®, Linux, MacOS®, otros), se decide programar la herramienta con los lenguajes de programación Java y Java 3D. El lenguaje Java se caracteriza por ser una lenguaje de programación y a la vez una plataforma; que se caracteriza por su sencillez, ser distribuido, dinámico, multiplataforma, robusto, de alto rendimiento entre otros. (Arismendi, Aguirre, & Barrios, 2008). Arismendi, citando a Bouvier dice: “Java 3D es una Interfaz de Programación de Aplicaciones (API) para generar e interactuar con gráficos tridimensionales utilizando el lenguaje de programación Java. Esta interfaz es una extensión estándar al equipo de herramientas de desarrollo en Java (JDK). El API provee un conjunto de conceptos, para la creación y manipulación de geometrías tridimensionales, y estructuras, para generar imágenes a partir de esas geometrías”.(Bouvier, 2002)

La herramienta computacional para la simulación utiliza las ecuaciones de la cinemática directa y de la cinemática inversa del robot manipulador propuesto, y calculadas en el presente trabajo. Además hace uso de un modelo tridimensional simplificado del robot manipulador, que se puede observar en la Figura 83, donde solamente se muestra la estructura del manipulador.

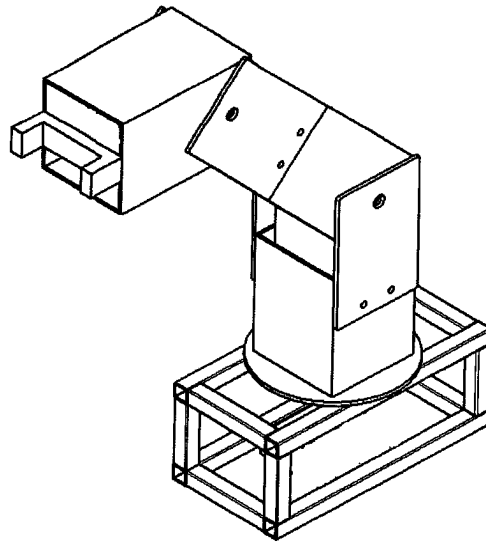


Figura 83 Modelo 3D simplificado del manipulador robótico

A continuación se muestra la interfaz de la herramienta computacional desarrollada, donde se explica las funciones que puede realizar. En la Figura 84 se muestra la interfaz completa del programa, donde se enseñan los elementos que la componen.

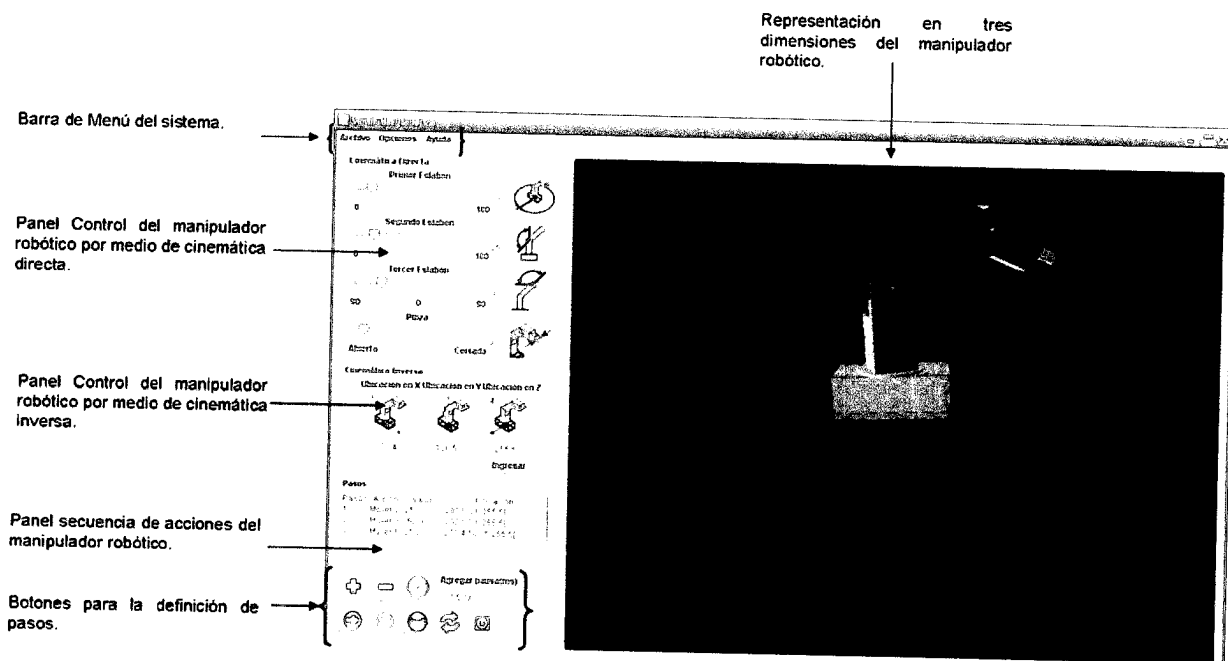


Figura 84 Interfaz de la herramienta computacional.
Fuente: (Arismendi, Aguirre, & Barrios, 2008)

Se tiene una Barra de Menú del sistema, donde se controlan las opciones del programa como abrir, guardar, cerrar, propiedades de visualización, ayuda, etc. También se encuentra en la interfaz el panel de control, que se divide en el control del a cinemática directa, ver Figura 85; y de la cinemática inversa, ver Figura 86.

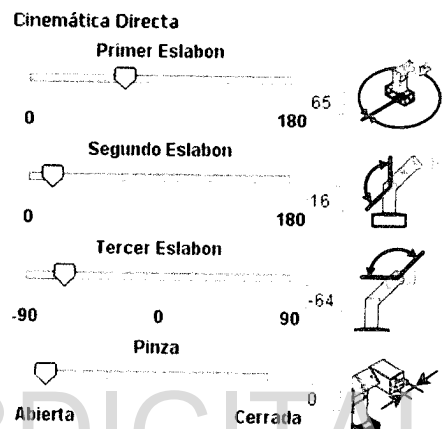


Figura 85 Panel de la cinemática directa.
Fuente: (Arismendi, Aguirre, & Barrios, 2008)

Cinemática Inversa

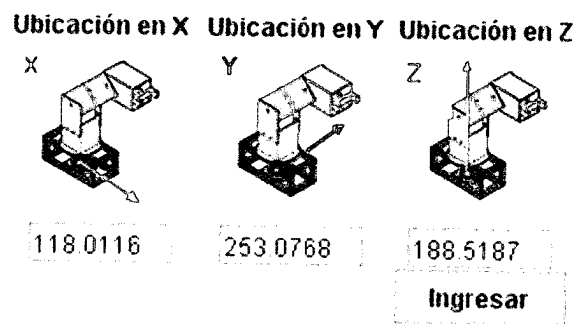


Figura 86 Panel de la cinemática inversa.
Fuente: (Arismendi, Aguirre, & Barrios, 2008)

En el panel de la cinemática directa, se pueden mover cada uno de los eslabones por separado, este movimiento se puede realizar introduciendo los valores articulares directamente o deslizando la barra de desplazamiento para cada eslabón, adicionalmente, al programa se le colocó una pinza en el eslabón número tres, esta pinza tiene dos posiciones, abierta y cerrada. Al lado de cada control de movimiento, se aprecia una figura que además de mostrar el eslabón que se mueve, indica como realiza dicho movimiento. El panel de la cinemática inversa tiene tres campos, donde se introducen las coordenadas del extremo del robot manipulador por separado, X, Y, Z. Sobre cada campo se tiene una figura que indica el sentido positivo para cada una de las coordenadas.

La introducción de datos se puede realizar indistintamente en cualquiera de los dos paneles. Si la introducción de datos se hace en el panel de la cinemática directa, los valores del panel de la cinemática inversa se actualizan automáticamente, y de igual manera, si se introducen los datos en el panel de la cinemática inversa, se actualizan automáticamente en el panel de la cinemática directa. Esta herramienta sirve para confirmar los resultados obtenidos anteriormente.

Luego se tiene un panel llamado PASOS, ver Figura 87, para la programación de tareas del manipulador. En este panel se agregan posiciones que se desea que tome el manipulador, el orden deseado, y pausas entre la ejecución de los movimientos. Una vez guardados los pasos que se quiere que realice el manipulador, se puede ejecutar la secuencia de movimientos del mismo; a medida que se ejecuta la secuencia, se va resaltando en la tabla en que paso está. Estas secuencias se pueden guardar en un archivo para ser utilizadas posteriormente.

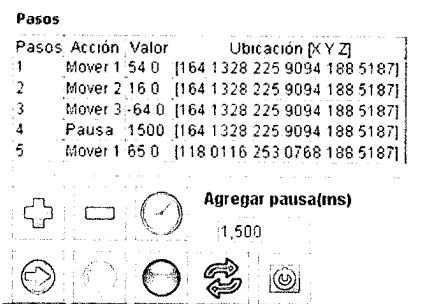


Figura 87 Panel de Pasos.

Fuente: (Arismendi, Aguirre, & Barrios, 2008)

Por último se tiene el panel gráfico, ver Figura 88, donde se muestra el modelo tridimensional del manipulador, una base (piso) que sirve como referencia y ubicación del manipulador en el espacio, y también se puede observar un ícono que indica los sentidos positivos del sistema de referencia. En este panel gráfico se puede observar en tiempo real los movimientos introducidos en los paneles de la cinemática directa y/o inversa, así como también la ejecución de las secuencias programadas en el panel de pasos.



Figura 88 Panel gráfico.

Fuente: (Arismendi, Aguirre, & Barrios, 2008)

El alcance del proyecto llegó hasta la programación de la cinemática directa, la cinemática inversa y la programación de secuencias en el simulador. La comunicación del simulador con el prototipo físico del manipulador no se pudo realizar, ya que para la fecha de culminación del proyecto, el manipulador no estaba totalmente construido. Adicionalmente, se hace la recomendación para trabajos futuros realizar la comunicación entre el simulador y el prototipo físico, así como la programación de las matrices Jacobianas y la dinámica del manipulador.

4.5 Costos del robot manipulador propuesto

Un aspecto importante en el presente trabajo es el bajo costo de fabricación del robot manipulador. Esto se debe al tamaño utilizado, que por ser de pequeñas dimensiones se reduce la cantidad de material a utilizar; estos materiales utilizados son en su mayoría piezas estándares modificadas para ser adaptadas al diseño elaborado, como el caso de los perfiles cuadrados de aluminio. En la Tabla 17 se presentan los costos del manipulador robótico del presente trabajo.

Tabla 17 Costos de fabricación del robot manipulador

Material y/o proceso de fabricación:	Precio
Eslabones uno, dos y tres	150 Bs.F
Base	140 Bs.F
4 motores LEGO ® NXT	170 Bs.F
Tornillos y tuercas	15 Bs.F
3 Ejes	60 Bs.F
Piezas varias de LEGO ®	45 Bs.F
Cable para conexión de motores	10 Bs.F
Total	590 Bs.F

Para mostrar la proporción de la reducción de los costos, se buscan precios de varios robots manipuladores y se comparan con el fabricado. En la Tabla 18 se muestran los robots manipuladores y sus precios.

Tabla 18 Comparación de precios de robots manipuladores

Robot manipulador	Precio
ABB IRB 140	\$ 40.000 = Bs.F. 86.000
KUKA KR3	€ 30.000 = Bs.F. 96.750
EduBot (educativo)	£ 1080 = Bs.F. 4.700
Robot manipulador propuesto	Bs.F. 590

Como se puede observar de la tabla anterior, el precio del robot manipulador propuesto es mucho menor que robots manipuladores comerciales. Y si se compara con un robot manipulador también para fines didácticos se observa que su precio puede llegar a ser ocho veces menor, sin incluir el precio por el envío y la importación.

WWW.BDIGITAL.ULA.VE

Licencia Creative Commons:
Atribución - No Comercial - Compartir Igual 3.0 Venezuela
(CC BY-NC-SA 3.0 VE)

Capítulo 5:

Conclusiones, recomendaciones y futuros trabajos

El presente capítulo muestra las conclusiones obtenidas luego de finalizado y analizado el diseño del robot manipulador. Se realizan también recomendaciones en caso de buscar mejorar el presente trabajo o de continuarlo. Se sugieren algunas alternativas en caso de querer continuar con el presente trabajo.

5.1 Conclusiones

Una vez finalizado el diseño y fabricación del robot manipulador propuesto, y analizados el proceso y los resultados obtenidos se llega a las conclusiones que se diseñó y construyó un prototipo de robot manipulador con tres grados de libertad para fines educativos. El mismo puede ser utilizado para realizar prácticas de programación de robots manipuladores, análisis de mecanismos, uso de sensores y actuadores, control de procesos. También puede ser utilizado para la investigación en el área de robótica industrial, como diseño de controladores específicos para tareas específicas, tele-operación, colaboración con otros robots, etc. Con esto se cumple el objetivo principal del presente trabajo.

El diseño y fabricación es realizado completamente en el país, demostrando que se tienen los recursos humanos necesarios para el desarrollo de este tipo de tecnologías. Aunque es de notar que el país carece del sector industrial necesario para este tipo de actividad, por lo que hay que importar gran cantidad de piezas y materiales, tal es el caso de los motores y sensores.

Se obtiene una notable reducción de costos en comparación con robots manipuladores comerciales y para la investigación. Por lo que la fabricación es una salida bastante viable para muchas de las universidades e institutos que tienen líneas de investigación en estas áreas. Así como también los dedicados a la enseñanza y entrenamiento.

El que se realice un proyecto de tesis de pre-grado que consiste desarrollar una herramienta computacional para la simulación y programación del robot manipulador propuesto en este proyecto, muestra que el mismo cumple con los objetivos de servir para fines educativos y de investigación en el área de robótica, en este caso en específicos en robótica de manipuladores industriales.

El uso de herramientas CAD es de gran ayuda para el diseño geométrico de robots manipuladores, permitiendo la evaluación y verificación de distintos aspectos del mismo antes de la fabricación, reduciendo así tiempos de fabricación y la necesidad de mayor cantidad de prototipos físicos, lo que se traduce en menos costos en el desarrollo del proyecto completo.

El uso de herramientas computacionales para el cálculo matemático también ayudó a la reducción considerable del tiempo de desarrollo del manipulador, permitiendo realizar simulaciones y variaciones de varias propuestas, casi en tiempo real, hasta que se llegar al diseño final del robot manipulador.

5.2 Recomendaciones

Para el mejoramiento del presente trabajo se pueden realizar las siguientes recomendaciones:

- Si se va a utilizar el manipulador robótico para el manejo de cargas, se recomienda realizar un análisis estructural del mismo, para evaluar su comportamiento y realizar arreglos a la estructura de ser necesario.
- De la misma manera, si se le añade un elemento terminal para añadir grados de libertad y agregarle orientación a las tareas que realiza el robot, realizar el análisis estructural para evaluar su comportamiento y realizar las correcciones necesarias.

5.3 Futuros trabajos

Al ser el comienzo de una línea de investigación, del presente trabajo surgen otros temas de investigación, entre los que se pueden sugerir:

- Agregar más grados de libertad al robot manipulador que permitan la orientación de las tareas a realizar.
- El análisis estructural del robot manipulador en caso de agregarle un elemento terminal o trabajar con manipulación de cargas.
- Comunicación entre el robot manipulador y la herramienta computacional de simulación y programación.
- Programación de distintos algoritmos de control para velocidad, aceleración, seguimiento de trayectorias.

BIBLIOGRAFÍA

Aguirre, I., & Lischinsky, P. (2007). *Aplicaciones de Robots*. Mérida: Universidad de Los Andes.

Arismendi, C., Aguirre, I., & Barrios, J. (2008). *Desarrollo de una Herramienta Computacional para la Simulación y Programación de un Manipulador Robótico de Tres Grados de Libertad*. Mérida: Universidad de Los Andes.

Autodesk. (2008). *Autodesk Inventor*. Recuperado el 05 de Febrero de 2008, de <http://usa.autodesk.com/adsk/servlet/index?siteID=123112&id=4246282>

Barrientos, A., Peñín, L. F., Balaguer, C., & Aracil, R. (1997). *Fundamentos de Robótica* (1era ed.). Madrid, España: McGraw-Hill.

Barrientos, A., Peñín, L. F., Balaguer, C., & Aracil, R. (2007). *Fundamentos de Robótica* (2da ed.). Madrid, España: McGraw-Hill.

Bermúdez, L., Borrero, A., & Lischinsky, P. *Aplicaciones de Robots en la industria*. Mérida: Universidad de Los Andes.

Bouvier, D. (2002). *Getting started with Java 3D API*. California, Estados Unidos: Sun Microsystems.

Cadenas, C., & Mora, P. O. (2001). *Diseño y construcción de un robot scara controlado por PLC*. Mérida: Universidad de Los Andes.

Cañizales Ponte, A. F., & Dulhoste, J. C. (1993). *Estudio teórico de un robot de dos grados de libertad con flexibilidad*. Mérida: Universidad de Los Andes.

Corke, P. I. (2006). *Robotic Toolbox for MATLAB*. Recuperado el 03 de Diciembre de 2007, de <http://www.petercorke.com/Robotics%20Toolbox.html>

Craig, J. J. (2006). *Robótica* (Tercera edición ed.). México: Pearson - Prentice Hall.

Herrera, R., & Lischinsky, P. (1998). *Desarrollo de una librería de modelos de brazos manipuladores y evaluación de leyes de control*. Mérida: Universidad de Los Andes.

LEGO (R). (2006). *Shop LEGO*. Recuperado el 12 de Febrero de 2008, de <http://shop.lego.com/ByCategory/Product.aspx?p=9842&cn=389&d=292>

LEGO Education. (22 de Febrero de 2006). *Mindstorms Education NXT*. Recuperado el 12 de Febrero de 2008, de Get Excited about NXT: Driving Straight Lines with Ease: <http://www.legoeducation.info/nxt/2006/02/22/get-excited-about-nxt-driving-straight-lines-with-ease/>

MapleSoft. (22 de Octubre de 2007). Recuperado el 22 de Octubre de 2007, de <http://www.maplesoft.com/products/Maple11/professionals/main.aspx>

MathWorks. (07 de Noviembre de 2007). *The MathWorks accelerating the pace of engineering and science*. Recuperado el 07 de Noviembre de 2007, de <http://www.mathworks.es/products/matlab/>

Real Academia Española. (2007). *Diccionario de la lengua española*. Recuperado el Septiembre de 2007, de <http://www.rae.es>

Torres, F., Pomares, J., Gil, P., Puente, S. T., & Aracil, R. (2002). *Robots y Sistemas Sensoriales* (2da ed.). Madrid, España: Pearson Educación S.A.

Valera Briceño, J. L. (1992). *Diseño de un control adaptativo para un robot manipulador con tres grados de libertad*. Mérida: Universidad de Los Andes.

Wikipedia. (21 de Enero de 2008). *Wikipedia - Autodesk Inventor*. Recuperado el 05 de Febrero de 2008, de http://en.wikipedia.org/wiki/Autodesk_Inventor

Wikipedia. (10 de Abril de 2006). *Wikipedia - Motor paso a paso*. Recuperado el 15 de Febrero de 2008, de http://es.wikipedia.org/wiki/Motor_paso_a_paso

Wikipedia. (16 de Enero de 2006). *Wikipedia - Servomotor*. Recuperado el 12 de Febrero de 2008, de <http://es.wikipedia.org/wiki/Servomotor>

WWW.BDIGITAL.ULA.VE

Anexo A

Programa elaborado en MAPLE ® para el cálculo de las cinemáticas directa e inversas, cálculo de las Jacobianas directa e inversa.

WWW.BDIGITAL.ULA.VE

Licencia Creative Commons:
Atribución - No Comercial - Compartir Igual 3.0 Venezuela
(CC BY-NC-SA 3.0 VE)

```
> restart.
```

```
> with(LinearAlgebra):
```

Definición de las Matrices de Rotación

```
> Td:=<<Nxd | Oxd | Axd | Pxd>, <Nyd | Oyd | Ayd | Pyd>, <Nzd | Ozd |
Azd | Pzd>, <0 | 0 | 0 | 1>>;
```

$$Td := \begin{bmatrix} Nxd & Oxd & Axd & Pxd \\ Nyd & Oyd & Ayd & Pyd \\ Nzd & Ozd & Azd & Pzd \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

```
> A01:=<<cos(theta1) | 0 | sin(theta1) | 0>, <sin(theta1) | 0 | -
cos(theta1) | 0>, <0 | 1 | 0 | 230>, <0 | 0 | 0 | 1>>;
```

$$A01 := \begin{bmatrix} \cos(\theta_1) & 0 & \sin(\theta_1) & 0 \\ \sin(\theta_1) & 0 & -\cos(\theta_1) & 0 \\ 0 & 1 & 0 & 230 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

```
> A12:=<<cos(theta2) | -sin(theta2) | 0 | 200*cos(theta2)>,
<sin(theta2) | cos(theta2) | 0 | 200*sin(theta2)>, <0 | 0 | 1 | 0>, <0
| 0 | 0 | 1>>;
```

$$A12 := \begin{bmatrix} \cos(\theta_2) & -\sin(\theta_2) & 0 & 200 \cos(\theta_2) \\ \sin(\theta_2) & \cos(\theta_2) & 0 & 200 \sin(\theta_2) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

```
> A23:=<<cos(theta3) | -sin(theta3) | 0 | 130*cos(theta3)>,
<sin(theta3) | cos(theta3) | 0 | 130*sin(theta3)>, <0 | 0 | 1 | 0>, <0
| 0 | 0 | 1>>;
```

$$A23 := \begin{bmatrix} \cos(\theta_3) & -\sin(\theta_3) & 0 & 130 \cos(\theta_3) \\ \sin(\theta_3) & \cos(\theta_3) & 0 & 130 \sin(\theta_3) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Cálculo de la cinemática directa

```
> Tdirecta:=A01.A12.A23.
```

$$Tdirecta := [[\cos(\theta_1) \cos(\theta_2) \cos(\theta_3) - \cos(\theta_1) \sin(\theta_2) \sin(\theta_3), -\cos(\theta_1) \cos(\theta_2) \sin(\theta_3) - \cos(\theta_1) \sin(\theta_2) \cos(\theta_3), \sin(\theta_1), 130 \cos(\theta_1) \cos(\theta_2) \cos(\theta_3) - 130 \cos(\theta_1) \sin(\theta_2) \sin(\theta_3) + 200 \cos(\theta_1) \cos(\theta_2)],$$

$$[\sin(\theta_1) \cos(\theta_2) \cos(\theta_3) - \sin(\theta_1) \sin(\theta_2) \sin(\theta_3), -\sin(\theta_1) \cos(\theta_2) \sin(\theta_3) - \sin(\theta_1) \sin(\theta_2) \cos(\theta_3), -\cos(\theta_1), 130 \sin(\theta_1) \cos(\theta_2) \cos(\theta_3) - 130 \sin(\theta_1) \sin(\theta_2) \sin(\theta_3) + 200 \sin(\theta_1) \cos(\theta_2)],$$

$$[\sin(\theta_2) \cos(\theta_3) + \cos(\theta_2) \sin(\theta_3), -\sin(\theta_2) \sin(\theta_3) + \cos(\theta_2) \cos(\theta_3), 0, 130 \sin(\theta_2) \cos(\theta_3) + 130 \cos(\theta_2) \sin(\theta_3) + 230 + 200 \sin(\theta_2)],$$

$$[0, 0, 0, 1]]$$

> Td[1,1]=Tdirecta[1,1];

$$Nxd = \cos(\theta_1) \cos(\theta_2) \cos(\theta_3) - \cos(\theta_1) \sin(\theta_2) \sin(\theta_3)$$

> Td[2,1]=Tdirecta[2,1];

$$Ny_d = \sin(\theta_1) \cos(\theta_2) \cos(\theta_3) - \sin(\theta_1) \sin(\theta_2) \sin(\theta_3)$$

> Td[3,1]=Tdirecta[3,1];

$$Nzd = \sin(\theta_2) \cos(\theta_3) + \cos(\theta_2) \sin(\theta_3)$$

> Td[1,2]=Tdirecta[1,2];

$$Oxd = -\cos(\theta_1) \cos(\theta_2) \sin(\theta_3) - \cos(\theta_1) \sin(\theta_2) \cos(\theta_3)$$

> Td[2,2]=Tdirecta[2,2];

$$Oyd = -\sin(\theta_1) \cos(\theta_2) \sin(\theta_3) - \sin(\theta_1) \sin(\theta_2) \cos(\theta_3)$$

> Td[3,2]=Tdirecta[3,2];

$$Ozd = -\sin(\theta_2) \sin(\theta_3) + \cos(\theta_2) \cos(\theta_3)$$

> Td[1,3]=Tdirecta[1,3];

$$Axd = \sin(\theta_1)$$

> Td[2,3]=Tdirecta[2,3];

$$Ayd = -\cos(\theta_1)$$

> Td[3,3]=Tdirecta[3,3];

$$Azd = 0$$

> Td[1,4]=Tdirecta[1,4];

$$Pxd = 130 \cos(\theta_1) \cos(\theta_2) \cos(\theta_3) - 130 \cos(\theta_1) \sin(\theta_2) \sin(\theta_3) + 200 \cos(\theta_1) \cos(\theta_2)$$

> Td[2,4]=Tdirecta[2,4];

$$Pzd = 130 \sin(\theta_1) \cos(\theta_2) \cos(\theta_3) - 130 \sin(\theta_1) \sin(\theta_2) \sin(\theta_3) + 200 \sin(\theta_1) \cos(\theta_2)$$

```
> Td[3,4]=Tdirecta[3,4];
```

$$Pzd = 130 \sin(\theta_2) \cos(\theta_3) + 130 \cos(\theta_2) \sin(\theta_3) + 230 + 200 \sin(\theta_2)$$

Definición de las Inversas de Las Matrices de Rotación

```
> A01inv:=<<A01[1,1] | A01[2,1] | A01[3,1] | (-Transpose(<A01[1,1] | A01[2,1] | A01[3,1]>).<A01[1,4], A01[2,4], A01[3,4]>)>, <A01[1,2] | A01[2,2] | A01[3,2] | (-Transpose(<A01[1,2] | A01[2,2] | A01[3,2]>).<A01[1,4], A01[2,4], A01[3,4]>)>, <A01[1,3] | A01[2,3] | A01[3,3] | (-Transpose(<A01[1,3] | A01[2,3] | A01[3,3]>).<A01[1,4], A01[2,4], A01[3,4]>)>, <0 | 0 | 0 | 1>>;
```

$$A01inv := \begin{bmatrix} \cos(\theta_1) & \sin(\theta_1) & 0 & 0 \\ 0 & 0 & 1 & -230 \\ \sin(\theta_1) & -\cos(\theta_1) & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

```
> A12inv:=<<A12[1,1] | A12[2,1] | A12[3,1] | (-Transpose(<A12[1,1] | A12[2,1] | A12[3,1]>).<A12[1,4], A12[2,4], A12[3,4]>)>, <A12[1,2] | A12[2,2] | A12[3,2] | (-Transpose(<A12[1,2] | A12[2,2] | A12[3,2]>).<A12[1,4], A12[2,4], A12[3,4]>)>, <A12[1,3] | A12[2,3] | A12[3,3] | (-Transpose(<A12[1,3] | A12[2,3] | A12[3,3]>).<A12[1,4], A12[2,4], A12[3,4]>)>, <0 | 0 | 0 | 1>>;
```

$$A12inv := \begin{bmatrix} \cos(\theta_2) & \sin(\theta_2) & 0 & -200 \cos(\theta_2)^2 - 200 \sin(\theta_2)^2 \\ -\sin(\theta_2) & \cos(\theta_2) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

```
> A12inv:=combine(A12inv, trig);
```

$$A12inv := \begin{bmatrix} \cos(\theta_2) & \sin(\theta_2) & 0 & -200 \\ -\sin(\theta_2) & \cos(\theta_2) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

```
> A23inv:=<<A23[1,1] | A23[2,1] | A23[3,1] | (-<A23[1,1] | A23[2,1] | A23[3,1]>.<A23[1,4], A23[2,4], A23[3,4]>)>, <A23[1,2] | A23[2,2] | A23[3,2] | (-<A23[1,2] | A23[2,2] | A23[3,2]>.<A23[1,4], A23[2,4], A23[3,4]>)>, <A23[1,3] | A23[2,3] | A23[3,3] | (-<A23[1,3] | A23[2,3] | A23[3,3]>.<A23[1,4], A23[2,4], A23[3,4]>)>, <0 | 0 | 0 | 1>>;
```

$$A23inv := \begin{bmatrix} \cos(\theta_3) & \sin(\theta_3) & 0 & -130 \cos(\theta_3)^2 - 130 \sin(\theta_3)^2 \\ -\sin(\theta_3) & \cos(\theta_3) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

```
> A23inv:=combine(A23inv, trig):
```

$$A23inv := \begin{bmatrix} \cos(\theta_3) & \sin(\theta_3) & 0 & -130 \\ -\sin(\theta_3) & \cos(\theta_3) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

```
> T:=<<Nx | Ox | Ax | Px>, <Ny | Oy | Ay | Py>, <Nz | Oz | Az | Pz>,
<0 | 0 | 0 | 1>>;
```

$$T := \begin{bmatrix} Nx & Ox & Ax & Px \\ Ny & Oy & Ay & Py \\ Nz & Oz & Az & Pz \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Cinemática Inversa

Para el cálculo de theta 1 se utiliza $A01inv \cdot T = A12 \cdot A23$ y de las 12 ecuaciones que quedan, se busca de tal manera que se pueda conseguir theta 1 en función de la posición final del manipulador (Px, Py, Pz)

```
> A01invT:= A01inv.T;
```

$$A01invT := \begin{bmatrix} \cos(\theta_1) Nx + \sin(\theta_1) Ny, \cos(\theta_1) Ox + \sin(\theta_1) Oy, \cos(\theta_1) Ax + \sin(\theta_1) Ay, \cos(\theta_1) Px + \sin(\theta_1) Py, \\ Nz, Oz, Az, -230 + Pz, \\ \sin(\theta_1) Nx - \cos(\theta_1) Ny, \sin(\theta_1) Ox - \cos(\theta_1) Oy, \sin(\theta_1) Ax - \cos(\theta_1) Ay, \\ \sin(\theta_1) Px - \cos(\theta_1) Py, \\ 0, 0, 0, 1 \end{bmatrix}$$

```
> A12A23:=simplify(A12.A23):
```

```
> MatrizA:=Matrix(3,4):
```

```
> for i from 1 by 1 to 3 do
```

```
>   for j from 1 by 1 to 4 do
```

```
>     MatrizA[i,j]:=simplify(A01invT[i,j]=A12A23[i,j]):
```

```
>     # MatrizA[i,j]:=combine(MatrizA[i,j], trig):
```

```
>   end do:
```

```
> end do:
```

```
> MatrizA[1,1]:
```

$$\cos(\theta_1) Nx + \sin(\theta_1) Ny = -\sin(\theta_2) \sin(\theta_3) + \cos(\theta_2) \cos(\theta_3)$$

```
> MatrizA[1,2]:
```

$$\cos(\theta_1) O_x + \sin(\theta_1) O_y = -\cos(\theta_2) \sin(\theta_3) - \sin(\theta_2) \cos(\theta_3)$$

> MatrizA[1,3];

$$\cos(\theta_1) A_x + \sin(\theta_1) A_y = 0$$

> MatrizA[1,4];

$$\cos(\theta_1) P_x + \sin(\theta_1) P_y = 130 \cos(\theta_2) \cos(\theta_3) - 130 \sin(\theta_2) \sin(\theta_3) + 200 \cos(\theta_2)$$

> MatrizA[2,1];

$$N_z = \sin(\theta_2) \cos(\theta_3) + \cos(\theta_2) \sin(\theta_3)$$

> MatrizA[2,2];

$$O_z = -\sin(\theta_2) \sin(\theta_3) + \cos(\theta_2) \cos(\theta_3)$$

> MatrizA[2,3];

$$A_z = 0$$

> MatrizA[2,4];

$$-230 + P_z = 130 \sin(\theta_2) \cos(\theta_3) + 130 \cos(\theta_2) \sin(\theta_3) + 200 \sin(\theta_2)$$

> MatrizA[3,1];

$$\sin(\theta_1) N_x - \cos(\theta_1) N_y = 0$$

> MatrizA[3,2];

$$\sin(\theta_1) O_x - \cos(\theta_1) O_y = 0$$

> MatrizA[3,3];

$$\sin(\theta_1) A_x - \cos(\theta_1) A_y = 1$$

> MatrizA[3,4];

$$\sin(\theta_1) P_x - \cos(\theta_1) P_y = 0$$

Cálculo de theta1

De las ecuaciones anteriores tomamos la celda A(3,4) y obtenemos que $\sin(\theta_1) P_x - \cos(\theta_1) P_y = 0$, por lo tanto $\theta_1 = \text{Arctg}(P_y/P_x)$

> isolate(MatrizA[3,4], theta1);

$$\theta_1 = \arctan\left(\frac{P_y}{P_x}\right)$$

> Teta1:=%;

$$Teta1 := \theta_1 = \arctan\left(\frac{P_y}{P_x}\right)$$

También de la misma ecuación MatrizA[3,4], tenemos que si elevamos ambos miembros de la igualdad al cuadrado llegamos a la ecuación

```
> A:=(sin(thetal)*Px-cos(thetal)*Py)^2;
```

$$A := (\sin(\theta 1) Px - \cos(\theta 1) Py)^2$$

```
> expand(A);
```

$$\sin(\theta 1)^2 Px^2 - 2 \sin(\theta 1) Px \cos(\theta 1) Py + \cos(\theta 1)^2 Py^2$$

Luego, despejando y aplicando más relaciones trigonométricas tendremos que:

```
> A:=cos(thetal)*Px+sin(thetal)*Py=sqrt(Px^2+Py^2);
```

$$A := \cos(\theta 1) Px + \sin(\theta 1) Py = \sqrt{Py^2 + Px^2}$$

Este resultado podría utilizarse más adelante para ayudar a simplificar otras ecuaciones.

Cálculo de theta2 y theta3

Para el cálculo de theta 2 y hteta3, se procede de igual manera que el caso anterior, utilizando $A12inv*A01inv*T = A23$

```
> A12invA01invT:=simplify(A12inv.A01inv.T);
```

```
> MatrizB:=Matrix(3,4);
```

```
> for i from 1 by 1 to 3 do
```

```
>   for j from 1 by 1 to 4 do
```

```
>     MatrizB[i,j]:=simplify(A12invA01invT[i,j]=A23[i,j]);
```

```
>     # MatrizB[i,j]:=combine(MatrizB[i,j], trig);
```

```
>   end do;
```

```
> end do;
```

```
> MatrizB[1,1];
```

$$\cos(\theta 1) \cos(\theta 2) Nx + \sin(\theta 1) \cos(\theta 2) Ny + \sin(\theta 2) Nz = \cos(\theta 3)$$

```
> MatrizB[1,2];
```

$$\cos(\theta 1) \cos(\theta 2) Ox + \sin(\theta 1) \cos(\theta 2) Oy + \sin(\theta 2) Oz = -\sin(\theta 3)$$

```
> MatrizB[1,3];
```

$$\cos(\theta 1) \cos(\theta 2) Ax + \sin(\theta 1) \cos(\theta 2) Ay + \sin(\theta 2) Az = 0$$

```
> MatrizB[1,4];
```

$$\begin{aligned} \cos(\theta_1) \cos(\theta_2) Px + \sin(\theta_1) \cos(\theta_2) Py + \sin(\theta_2) Pz - 200 - 230 \sin(\theta_2) \\ = 130 \cos(\theta_3) \end{aligned}$$

> MatrizB[2,1];

$$-\sin(\theta_2) \cos(\theta_1) Nx - \sin(\theta_2) \sin(\theta_1) Ny + \cos(\theta_2) Nz = \sin(\theta_3)$$

> MatrizB[2,2];

$$-\sin(\theta_2) \cos(\theta_1) Ox - \sin(\theta_2) \sin(\theta_1) Oy + \cos(\theta_2) Oz = \cos(\theta_3)$$

> MatrizB[2,3];

$$-\sin(\theta_2) \cos(\theta_1) Ax - \sin(\theta_2) \sin(\theta_1) Ay + \cos(\theta_2) Az = 0$$

> MatrizB[2,4];

$$-\sin(\theta_2) \cos(\theta_1) Px - \sin(\theta_2) \sin(\theta_1) Py + \cos(\theta_2) Pz - 230 \cos(\theta_2) = 130 \sin(\theta_3)$$

> MatrizB[3,1];

$$\sin(\theta_1) Nx - \cos(\theta_1) Ny = 0$$

> MatrizB[3,2];

$$\sin(\theta_1) Ox - \cos(\theta_1) Oy = 0$$

> MatrizB[3,3];

$$\sin(\theta_1) Ax - \cos(\theta_1) Ay = 1$$

> MatrizB[3,4];

$$\sin(\theta_1) Px - \cos(\theta_1) Py = 0$$

Del conjunto de ecuaciones anteriores tomamos las ecuaciones 45 (MatrizB[1,4]) y la ecuación 49 (Matriz[2,4]) para hallar las soluciones para theta2 y theta3

Primero tenemos las igualdades matemáticas siguientes:

$$a \cdot \cos(\theta) + b \cdot \sin(\theta) = C$$

$$u = \tan(\theta/2);$$

$$\cos(\theta) = (1 - u^2) / (1 + u^2)$$

$$\sin(\theta) = 2u / (1 + u^2)$$

por lo tanto:

$$a*(1-u^2)/(1+u^2) + b*(2*u)/(1+u^2) = c$$

$$a*(1-u^2) + b(2u) = c(1+u^2)$$

$$a - a*u^2 + 2*b*u = c + c*u^2$$

$$c+c*u^2-a+a*u^2-2*b*u = 0$$

$$u^2*(a+c) - 2*b*u + (c-a) = 0 \quad \% \text{ funcion cuadratica}$$

la ultima ecuación tiene como solución

$$u = (-(-2*b) \pm \sqrt{(-2*b)^2 - 4*(a+c)*(c-a)}) / (2*(a+c))$$

$$u = (b \pm \sqrt{b^2 + a^2 - c^2}) / (a+c)$$

y como $u = \tan(\theta/2)$, despejamos y tenemos θ

$$\theta = 2*\text{atan}(u)$$

Primero tomamos las ecuaciones 45 y 49 y agrupamos terminos

```
> Ec1:= MatrizB[1,4];
```

$$\begin{aligned} Ec1 &:= \cos(\theta_1) \cos(\theta_2) Px + \sin(\theta_1) \cos(\theta_2) Py + \sin(\theta_2) Pz - 200 - 230 \sin(\theta_2) \\ &= 130 \cos(\theta_3) \end{aligned}$$

```
> Ec2:= MatrizB[2,4];
```

$$\begin{aligned} Ec2 &:= -\sin(\theta_2) \cos(\theta_1) Px - \sin(\theta_2) \sin(\theta_1) Py + \cos(\theta_2) Pz - 230 \cos(\theta_2) \\ &= 130 \sin(\theta_3) \end{aligned}$$

```
> Ec1:= collect(lhs(Ec1), {cos(theta2), sin(theta2)})=rhs(Ec1);
```

$$\begin{aligned} Ec1 &:= (\cos(\theta_1) Px + \sin(\theta_1) Py) \cos(\theta_2) - 200 + (-230 + Pz) \sin(\theta_2) = 130 \cos \\ &(\theta_3) \end{aligned}$$

```
> Ec2:= collect(lhs(Ec2), {cos(theta2), sin(theta2)})=rhs(Ec2);
```

$$Ec2 := (-230 + Pz) \cos(\theta 2) + (-\cos(\theta 1) Px - \sin(\theta 1) Py) \sin(\theta 2) = 130 \sin(\theta 3)$$

Ahora simplificamos las ecuaciones para poder utilizar mas facilmente las igualdades mencionadas anteriormente

> A := cos(theta1)*Px+sin(theta1)*Py=a :

$$A := \cos(\theta 1) Px + \sin(\theta 1) Py = a$$

> B := -230+Pz=b ;

$$B := -230 + Pz = b$$

> Ec1 := algsubs(A, Ec1) ;

$$Ec1 := \cos(\theta 2) a - 200 + (-230 + Pz) \sin(\theta 2) = 130 \cos(\theta 3)$$

> Ec1 := algsubs(B, Ec1) ;

$$Ec1 := \sin(\theta 2) b - 200 + \cos(\theta 2) a = 130 \cos(\theta 3)$$

> Ec2 := algsubs(A, Ec2) ;

$$Ec2 := -\sin(\theta 2) a + (-230 + Pz) \cos(\theta 2) = 130 \sin(\theta 3)$$

> Ec2 := algsubs(B, Ec2) ;

$$Ec2 := \cos(\theta 2) b - \sin(\theta 2) a = 130 \sin(\theta 3)$$

Ahora para el calculo de theta3

> Ec3izq := lhs(Ec2)/lhs(Ec1) ;

$$Ec3izq := \frac{\cos(\theta 2) b - \sin(\theta 2) a}{\sin(\theta 2) b - 200 + \cos(\theta 2) a}$$

> Ec3der := rhs(Ec2)/rhs(Ec1) ;

$$Ec3der := \frac{\sin(\theta 3)}{\cos(\theta 3)}$$

> Ec3 := Ec3izq = Ec3der ;

$$Ec3 := \frac{\cos(\theta 2) b - \sin(\theta 2) a}{\sin(\theta 2) b - 200 + \cos(\theta 2) a} = \frac{\sin(\theta 3)}{\cos(\theta 3)}$$

> Ec3 := lhs(Ec3) = subs(sin(theta3)/cos(theta3)=tan(theta3), rhs(Ec3)) ;

$$Ec3 := \frac{\cos(\theta 2) b - \sin(\theta 2) a}{\sin(\theta 2) b - 200 + \cos(\theta 2) a} = \tan(\theta 3)$$

> Teta3 := isolate(Ec3, theta3) ;

$$Teta3 := \theta 3 = -\arctan\left(\frac{-\cos(\theta 2) b + \sin(\theta 2) a}{\sin(\theta 2) b - 200 + \cos(\theta 2) a}\right)$$

Ahora pra el cálculo de theta2

Primero elevamos al cuadrado ambos lados de las dos ecuaciones involucradas

Licencia Creative Commons:

Atribución - No Comercial - Compartir Igual 3.0 Venezuela
(CC BY-NC-SA 3.0 VE)

```
> Ec4der := rhs(Ec1)^2
```

$$Ec4der := 16900 \cos(\theta_3)^2$$

```
> Ec4izq := expand(lhs(Ec1)^2);
```

$$Ec4izq := \sin(\theta_2)^2 b^2 - 400 \sin(\theta_2) b + 2 \sin(\theta_2) b \cos(\theta_2) a + 40000 - 400 \cos(\theta_2) a + \cos(\theta_2)^2 a^2$$

```
> Ec4 := Ec4izq = Ec4der;
```

$$Ec4 := \sin(\theta_2)^2 b^2 - 400 \sin(\theta_2) b + 2 \sin(\theta_2) b \cos(\theta_2) a + 40000 - 400 \cos(\theta_2) a + \cos(\theta_2)^2 a^2 = 16900 \cos(\theta_3)^2$$

```
> Ec5der := rhs(Ec2)^2;
```

$$Ec5der := 16900 \sin(\theta_3)^2$$

```
> Ec5izq := expand(lhs(Ec2)^2);
```

$$Ec5izq := \cos(\theta_2)^2 b^2 - 2 \sin(\theta_2) b \cos(\theta_2) a + \sin(\theta_2)^2 a^2$$

```
> Ec5 := Ec5izq = Ec5der;
```

$$Ec5 := \cos(\theta_2)^2 b^2 - 2 \sin(\theta_2) b \cos(\theta_2) a + \sin(\theta_2)^2 a^2 = 16900 \sin(\theta_3)^2$$

Una vez que han sido elevadas al cuadrado, se suma las ecuaciones resultantes, a fin de simplificarlas.

```
> Ec6izq := Ec4izq + Ec5izq;
```

$$Ec6izq := \cos(\theta_2)^2 b^2 + \sin(\theta_2)^2 a^2 + \sin(\theta_2)^2 b^2 - 400 \sin(\theta_2) b + 40000 - 400 \cos(\theta_2) a + \cos(\theta_2)^2 a^2$$

```
> Ec6izq := simplify(Ec6izq, trig);
```

$$Ec6izq := 40000 - 400 \sin(\theta_2) b - 400 \cos(\theta_2) a + a^2 + b^2$$

```
> Ec6der := Ec4der + Ec5der;
```

$$Ec6der := 16900 \cos(\theta_3)^2 + 16900 \sin(\theta_3)^2$$

```
> Ec6der := simplify(Ec6der, trig);
```

$$Ec6der := 16900$$

```
> Ec6 := Ec6izq = Ec6der;
```

$$Ec6 := 40000 - 400 \sin(\theta_2) b - 400 \cos(\theta_2) a + a^2 + b^2 = 16900$$

```
> Ec6 := -rhs(Ec6) + lhs(Ec6) = 0;
```

$$Ec6 := 23100 - 400 \sin(\theta_2) b - 400 \cos(\theta_2) a + a^2 + b^2 = 0$$

```
> Ec6 := isolate(Ec6, 23100);
```

$$Ec6 := 23100 = 400 \sin(\theta_2) b + 400 \cos(\theta_2) a - a^2 - b^2$$

$$> Ec6 := (23100 + a^2 + b^2) / 400 = \sin(\theta_2) b + \cos(\theta_2) a;$$

$$Ec6 := \frac{231}{4} + \frac{1}{400} a^2 + \frac{1}{400} b^2 = \sin(\theta_2) b + \cos(\theta_2) a$$

$$> C := 231/4 + 1/400 * a^2 + 1/400 * b^2 = c;$$

$$C := \frac{231}{4} + \frac{1}{400} a^2 + \frac{1}{400} b^2 = c$$

Utilizando las igualdades:

$$a \cos(\theta) + b \sin(\theta) = C$$

$$u = \tan(\theta/2);$$

$$\cos(\theta) = (1 - u^2) / (1 + u^2)$$

$$\sin(\theta) = 2u / (1 + u^2)$$

por lo tanto:

$$a \frac{(1 - u^2)}{(1 + u^2)} + b \frac{(2u)}{(1 + u^2)} = c$$

$$a(1 - u^2) + b(2u) = c(1 + u^2)$$

$$a - a u^2 + 2 b u = c + c u^2$$

$$c + c u^2 - a + a u^2 - 2 b u = 0$$

$$u^2(a + c) - 2 b u + (c - a) = 0 \quad \% \text{funcion cuadratica}$$

la ultima ecuación tiene como solución

$$u = \frac{-(-2b) \pm \sqrt{(-2b)^2 - 4(a+c)(c-a)}}{2(a+c)}$$

$$u = \frac{(b \pm \sqrt{b^2 + a^2 - c^2})}{(a+c)}$$

y como $u = \tan(\theta/2)$, despejamos y tenemos θ

$$\theta = 2 \operatorname{atan}(u)$$

```
> Ua := (b + sqrt(b^2+a^2-c^2)) / (a+c) ;
```

$$Ua := \frac{b + \sqrt{b^2 + a^2 - c^2}}{a + c}$$

```
> Ub := (b - sqrt(b^2+a^2-c^2)) / (a+c) ;
```

$$Ub := \frac{b - \sqrt{b^2 + a^2 - c^2}}{a + c}$$

```
> Teta2a := 2*arctan(Ua) ;
```

$$Teta2a := 2 \arctan\left(\frac{b + \sqrt{b^2 + a^2 - c^2}}{a + c}\right)$$

```
> Teta2b := 2*arctan(Ub) ;
```

$$Teta2b := 2 \arctan\left(\frac{b - \sqrt{b^2 + a^2 - c^2}}{a + c}\right)$$

```
> C := rhs(C) = lhs(C) ;
```

$$C := c = \frac{231}{4} + \frac{1}{400} a^2 + \frac{1}{400} b^2$$

```
> B := rhs(B) = lhs(B) ;
```

$$B := b = -230 + Pz$$

```
> A := rhs(A) = lhs(A) ;
```

$$A := a = \cos(\theta) Px + \sin(\theta) Py$$

```
> Teta2a := algsubs(C, Teta2a) ;
```

$$Teta2a := 2 \arctan$$

$$\left(\frac{1}{a + \frac{231}{4} + \frac{1}{400} a^2 + \frac{1}{400} b^2} \right) (b$$

+

$$\sqrt{\frac{569}{800} b^2 - \frac{53361}{16} + \frac{569}{800} a^2 - \frac{1}{160000} a^4 - \frac{1}{80000} a^2 b^2 - \frac{1}{160000} b^4})$$

```
> Teta2a := algsubs(A, Teta2a) ;
```

$$\begin{aligned}
Teta2a := & 2 \arctan \left(\left(b \right. \right. \\
& + \left(-\frac{1}{40000} b^2 \cos(\theta) Px \sin(\theta) Py - \frac{1}{160000} \cos(\theta)^4 Px^4 \right. \\
& - \frac{1}{40000} \cos(\theta)^3 Px^3 \sin(\theta) Py - \frac{3}{80000} \cos(\theta)^2 Px^2 \sin(\theta)^2 Py^2 \\
& - \frac{1}{40000} \cos(\theta) Px \sin(\theta)^3 Py^3 - \frac{1}{160000} \sin(\theta)^4 Py^4 + \frac{569}{800} \cos(\theta) \\
& \left. \left. \right)^2 Px^2 - \frac{1}{80000} b^2 \cos(\theta)^2 Px^2 + \frac{569}{800} \sin(\theta)^2 Py^2 - \frac{1}{80000} b^2 \sin(\theta)^2 Py^2 \right. \\
& - \left. \frac{53361}{16} + \frac{569}{400} \sin(\theta) Px \cos(\theta) Py + \frac{569}{800} b^2 - \frac{1}{160000} b^4 \right)^{1/2} \left(\sin \right. \\
& \left. (\theta) Py + \frac{1}{400} \cos(\theta)^2 Px^2 + \frac{1}{200} \sin(\theta) Px \cos(\theta) Py + \frac{1}{400} \sin(\theta) \right. \\
& \left. \left. \right)^2 Py^2 + \frac{1}{400} b^2 + \cos(\theta) Px + \frac{231}{4} \right) \left. \right)
\end{aligned}$$

> Teta2a := algsubs(B, Teta2a);

$$\begin{aligned}
Teta2a := & 2 \arctan \left(\left(-230 + Pz \right. \right. \\
& + \left(\frac{23}{4000} \sin(\theta)^2 Py^2 Pz + 16800 - 23 Pz - \frac{509}{400} Pz^2 + \frac{23}{4000} Pz^3 \right. \\
& - \frac{1}{160000} Pz^4 + \frac{1}{20} \sin(\theta)^2 Py^2 - \frac{1}{40000} \cos(\theta)^3 Px^3 \sin(\theta) Py \\
& - \frac{3}{80000} \cos(\theta)^2 Px^2 \sin(\theta)^2 Py^2 + \frac{1}{20} \cos(\theta)^2 Px^2 + \frac{23}{4000} \cos(\theta) \\
& \left. \left. \right)^2 Px^2 Pz - \frac{1}{80000} \cos(\theta)^2 Px^2 Pz^2 + \frac{1}{10} \sin(\theta) Px \cos(\theta) Py + \frac{23}{2000} \cos \right. \\
& \left. (\theta) Px \sin(\theta) Py Pz - \frac{1}{40000} \cos(\theta) Px \sin(\theta) Py Pz^2 - \frac{1}{160000} \cos \right. \\
& \left. (\theta)^4 Px^4 - \frac{1}{80000} \sin(\theta)^2 Py^2 Pz^2 - \frac{1}{160000} \sin(\theta)^4 Py^4 - \frac{1}{40000} \cos(\theta) \right. \\
& \left. \left. \right) Px \sin(\theta)^3 Py^3 \right)^{1/2} \left(\frac{1}{400} \cos(\theta)^2 Px^2 + 190 - \frac{23}{20} Pz + \frac{1}{400} Pz^2 \right. \\
& + \frac{1}{400} \sin(\theta)^2 Py^2 + \sin(\theta) Py + \cos(\theta) Px + \frac{1}{200} \sin(\theta) Px \cos(\theta) \\
& \left. \left. \right) Py \right) \left. \right)
\end{aligned}$$

> Teta3 := algsubs(A, Teta3);

$$\begin{aligned}
Teta3 := & \theta3 = -\arctan \\
& \left(\frac{-\cos(\theta2) b + \sin(\theta2) \cos(\theta) Px + \sin(\theta2) \sin(\theta) Py}{\sin(\theta2) b - 200 + \cos(\theta) \cos(\theta2) Px + \sin(\theta) \cos(\theta2) Py} \right)
\end{aligned}$$

> Teta3 := algsubs(B, Teta3);

$Teta3 := \theta3 = -\arctan$

$$\left(\frac{\sin(\theta2) \cos(\theta1) Px + \sin(\theta2) \sin(\theta1) Py - \cos(\theta2) Pz + 230 \cos(\theta2)}{\cos(\theta1) \cos(\theta2) Px + \sin(\theta1) \cos(\theta2) Py + \sin(\theta2) Pz - 200 - 230 \sin(\theta2)} \right)$$

**

**

JACOBIANAS

**

**

JACOBIANA DIRECTA

> Jdirecta := Matrix(3,3):

> Jdirecta[1,1] := diff(Tdirecta[1,4], theta1);

$$Jdirecta_{1,1} := -130 \sin(\theta1) \cos(\theta2) \cos(\theta3) + 130 \sin(\theta1) \sin(\theta2) \sin(\theta3) - 200 \sin(\theta1) \cos(\theta2)$$

> Jdirecta[1,2] := diff(Tdirecta[1,4], theta2);

$$Jdirecta_{1,2} := -130 \cos(\theta1) \sin(\theta2) \cos(\theta3) - 130 \cos(\theta1) \cos(\theta2) \sin(\theta3) - 200 \sin(\theta2) \cos(\theta1)$$

> Jdirecta[1,3] := diff(Tdirecta[1,4], theta3);

$$Jdirecta_{1,3} := -130 \cos(\theta1) \cos(\theta2) \sin(\theta3) - 130 \cos(\theta1) \sin(\theta2) \cos(\theta3)$$

> Jdirecta[2,1] := diff(Tdirecta[2,4], theta1);

$$Jdirecta_{2,1} := 130 \cos(\theta1) \cos(\theta2) \cos(\theta3) - 130 \cos(\theta1) \sin(\theta2) \sin(\theta3) + 200 \cos(\theta1) \cos(\theta2)$$

> Jdirecta[2,2] := diff(Tdirecta[2,4], theta2);

$$Jdirecta_{2,2} := -130 \sin(\theta1) \sin(\theta2) \cos(\theta3) - 130 \sin(\theta1) \cos(\theta2) \sin(\theta3) - 200 \sin(\theta2) \sin(\theta1)$$

> Jdirecta[2,3] := diff(Tdirecta[2,4], theta3);

Licencia Creative Commons:

Atribución - No Comercial - Compartir Igual 3.0 Venezuela
(CC BY-NC-SA 3.0 VE)

```

Jdirecta2,3 := -130 sin(θ1) cos(θ2) sin(θ3) - 130 sin(θ1) sin(θ2) cos(θ3)
> Jdirecta[3,1] := diff(Tdirecta[3,4], theta1);

Jdirecta3,1 := 0
> Jdirecta[3,2] := diff(Tdirecta[3,4], theta2);

Jdirecta3,2 := 130 cos(θ2) cos(θ3) - 130 sin(θ2) sin(θ3) + 200 cos(θ2)
> Jdirecta[3,3] := diff(Tdirecta[3,4], theta3);

Jdirecta3,3 := -130 sin(θ2) sin(θ3) + 130 cos(θ2) cos(θ3)
> Jdirecta:

```

```

[[[-130 sin(θ1) cos(θ2) cos(θ3) + 130 sin(θ1) sin(θ2) sin(θ3) - 200 sin(θ1) cos
(θ2), -130 cos(θ1) sin(θ2) cos(θ3) - 130 cos(θ1) cos(θ2) sin(θ3) - 200 sin
(θ2) cos(θ1), -130 cos(θ1) cos(θ2) sin(θ3) - 130 cos(θ1) sin(θ2) cos(θ3)],
[130 cos(θ1) cos(θ2) cos(θ3) - 130 cos(θ1) sin(θ2) sin(θ3) + 200 cos(θ1)
) cos(θ2), -130 sin(θ1) sin(θ2) cos(θ3) - 130 sin(θ1) cos(θ2) sin(θ3)
- 200 sin(θ2) sin(θ1), -130 sin(θ1) cos(θ2) sin(θ3) - 130 sin(θ1) sin(θ2
) cos(θ3)],
[0, 130 cos(θ2) cos(θ3) - 130 sin(θ2) sin(θ3) + 200 cos(θ2), -130 sin(θ2) sin
(θ3) + 130 cos(θ2) cos(θ3)]]]

```

JACOBIANA INVERSA DIRECTA - Invirtiendo la jacobiana directa

```

> JinversaDir := simplify(MatrixInverse(Jdirecta));

```

$$\begin{aligned}
 JinversaDir := & \left[\left[\frac{1}{10} \frac{\sin(\theta_1)}{-20 \cos(\theta_2) - 13 \cos(\theta_2) \cos(\theta_3) + 13 \sin(\theta_2) \sin(\theta_3)}, \right. \right. \\
 & \left. \left. -\frac{1}{10} \frac{\cos(\theta_1)}{-20 \cos(\theta_2) - 13 \cos(\theta_2) \cos(\theta_3) + 13 \sin(\theta_2) \sin(\theta_3)}, 0 \right], \right. \\
 & \left[-\frac{1}{200} \frac{(\sin(\theta_2) \sin(\theta_3) - \cos(\theta_2) \cos(\theta_3)) \cos(\theta_1)}{\sin(\theta_3)}, \right. \\
 & \left. -\frac{1}{200} \frac{(\sin(\theta_2) \sin(\theta_3) - \cos(\theta_2) \cos(\theta_3)) \sin(\theta_1)}{\sin(\theta_3)}, \right. \\
 & \left. \frac{1}{200} \frac{\sin(\theta_2) \cos(\theta_3) + \cos(\theta_2) \sin(\theta_3)}{\sin(\theta_3)} \right], \\
 & \left[\frac{1}{2600} \frac{\cos(\theta_1) (-20 \cos(\theta_2) - 13 \cos(\theta_2) \cos(\theta_3) + 13 \sin(\theta_2) \sin(\theta_3))}{\sin(\theta_3)}, \right. \\
 & \left. \frac{1}{2600} \frac{\sin(\theta_1) (-20 \cos(\theta_2) - 13 \cos(\theta_2) \cos(\theta_3) + 13 \sin(\theta_2) \sin(\theta_3))}{\sin(\theta_3)}, \right. \\
 & \left. \left. -\frac{1}{2600} \frac{13 \sin(\theta_2) \cos(\theta_3) + 20 \sin(\theta_2) + 13 \cos(\theta_2) \sin(\theta_3)}{\sin(\theta_3)} \right] \right]
 \end{aligned}$$

JACOBIANA INVERSA - Derivando la cinemática inversa

```

> Jinversa := Matrix(3,3):

```

Licencia Creative Commons:

Atribución - No Comercial - Compartir Igual 3.0 Venezuela
(CC BY-NC-SA 3.0 VE)

```
> Jinversa[1,1]:= diff(rhs(Teta1), Px);
```

$$Jinversa_{1,1} := -\frac{Py}{Px^2 \left(1 + \frac{Py^2}{Px^2}\right)}$$

```
> Jinversa[1,2]:= diff(rhs(Teta1), Py);
```

$$Jinversa_{1,2} := \frac{1}{Px \left(1 + \frac{Py^2}{Px^2}\right)}$$

```
> Jinversa[1,3]:= diff(rhs(Teta1), Pz);
```

$$Jinversa_{1,3} := 0$$

```
> Jinversa[2,1]:= simplify(diff(Teta2a, Px));
```

```
> Jinversa[2,2]:= simplify(diff(Teta2a, Py));
```

```
> Jinversa[2,3]:= simplify(diff(Teta2a, Pz));
```

```
> Jinversa[3,1]:= simplify(diff(rhs(Teta3), Px));
```

$$Jinversa_{3,1} := -(\cos(\theta 1) (-200 \sin(\theta 2) - 230 + Pz)) / (92900 - 400 \sin(\theta 1) \cos(\theta 2) Py + \cos(\theta 1)^2 Px^2 - 400 \sin(\theta 2) Pz - 400 \cos(\theta 1) \cos(\theta 2) Px + 2 \sin(\theta 1) Px \cos(\theta 1) Py + Py^2 - \cos(\theta 1)^2 Py^2 - 460 Pz + Pz^2 + 92000 \sin(\theta 2))$$

```
> Jinversa[3,2]:= simplify(diff(rhs(Teta3), Py));
```

$$Jinversa_{3,2} := -(\sin(\theta 1) (-200 \sin(\theta 2) - 230 + Pz)) / (92900 - 400 \sin(\theta 1) \cos(\theta 2) Py + \cos(\theta 1)^2 Px^2 - 400 \sin(\theta 2) Pz - 400 \cos(\theta 1) \cos(\theta 2) Px + 2 \sin(\theta 1) Px \cos(\theta 1) Py + Py^2 - \cos(\theta 1)^2 Py^2 - 460 Pz + Pz^2 + 92000 \sin(\theta 2))$$

```
> Jinversa[3,3]:= simplify(diff(rhs(Teta3), Pz));
```

$$Jinversa_{3,3} := (-200 \cos(\theta 2) + \cos(\theta 1) Px + \sin(\theta 1) Py) / (92900 - 400 \sin(\theta 1) \cos(\theta 2) Py + \cos(\theta 1)^2 Px^2 - 400 \sin(\theta 2) Pz - 400 \cos(\theta 1) \cos(\theta 2) Px + 2 \sin(\theta 1) Px \cos(\theta 1) Py + Py^2 - \cos(\theta 1)^2 Py^2 - 460 Pz + Pz^2 + 92000 \sin(\theta 2))$$

```
>
```

Anexo B

Programa elaborado en MAPLE para el cálculo de dinámica del manipulador, siguiendo el algoritmo de Newton – Euler.

WWW.BDIGITAL.ULA.VE

Licencia Creative Commons:
Atribución - No Comercial - Compartir Igual 3.0 Venezuela
(CC BY-NC-SA 3.0 VE)

```
> restart;
```

```
> with(LinearAlgebra):
```

Calculo de la Dinámica del manipulador por medio de la formulación de Newton-Euler

Definición de las Matrices de Rotación

Para este caso las matrices de rotación son de 3x3

```
> A01:=<<cos(theta1) | 0 | sin(theta1)>, <sin(theta1) | 0 | -cos(theta1)>, <0 | 1 | 0>>;
```

$$A_{01} := \begin{pmatrix} \cos(\theta_1) & 0 & \sin(\theta_1) \\ \sin(\theta_1) & 0 & -\cos(\theta_1) \\ 0 & 1 & 0 \end{pmatrix}$$

```
> A12:=<<cos(theta2) | -sin(theta2) | 0>, <sin(theta2) | cos(theta2) | 0>, <0 | 0 | 1>>;
```

$$A_{12} := \begin{pmatrix} \cos(\theta_2) & -\sin(\theta_2) & 0 \\ \sin(\theta_2) & \cos(\theta_2) & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

```
> A23:=<<cos(theta3) | -sin(theta3) | 0>, <sin(theta3) | cos(theta3) | 0>, <0 | 0 | 1>>;
```

$$A_{23} := \begin{pmatrix} \cos(\theta_3) & -\sin(\theta_3) & 0 \\ \sin(\theta_3) & \cos(\theta_3) & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

```
> A01inv := combine(MatrixInverse(A01), trig);
```

$$A_{01inv} := \begin{pmatrix} \cos(\theta_1) & \sin(\theta_1) & 0 \\ 0 & 0 & 1 \\ \sin(\theta_1) & -\cos(\theta_1) & 0 \end{pmatrix}$$

```
> A12inv := combine(MatrixInverse(A12), trig);
```

$$A_{12inv} := \begin{pmatrix} \cos(\theta_2) & \sin(\theta_2) & 0 \\ -\sin(\theta_2) & \cos(\theta_2) & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

```
> A23inv := combine(MatrixInverse(A23), trig);
```

$$A_{23inv} := \begin{pmatrix} \cos(\theta_3) & \sin(\theta_3) & 0 \\ -\sin(\theta_3) & \cos(\theta_3) & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Ahora se establecen las **condiciones iniciales**

```
> VelAngIni := < 0, 0, 0 >;
```

$$VelAngIni := \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$$

```
> AceAngIni := < 0, 0, 0 >;
```

$$AceAngI_i := \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

$$> VelLinI_i := \langle 0, 0, 0 \rangle;$$

$$VelLinI_i := \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

$$> AceLinI_i := \langle 0, 0, gz \rangle;$$

$$AceLinI_i := \begin{bmatrix} 0 \\ 0 \\ gz \end{bmatrix}$$

$$> Zo := \langle 0, 0, 1 \rangle;$$

$$Zo := \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

Para el calculo inicial se supondrá que no se ejerce ninguna fuerza ni ningun par sobre el extremo del robot.

Ahora se calculan las **coordenadas de cada eslabón con respecto al eslabón anterior**. = $[a_i, d_i S_i, d_i C_i]$

$$> P11 := \langle 0, 230, 0 \rangle;$$

$$P11 := \begin{bmatrix} 0 \\ 230 \\ 0 \end{bmatrix}$$

$$> P22 := \langle 200, 0, 0 \rangle;$$

$$P22 := \begin{bmatrix} 200 \\ 0 \\ 0 \end{bmatrix}$$

$$> P33 := \langle 130, 0, 0 \rangle;$$

$$P33 := \begin{bmatrix} 130 \\ 0 \\ 0 \end{bmatrix}$$

Coordenadas del centro de masas del eslabón i con respecto al sistema Si

$$> S11 := \langle -2.807, -82.111, 0 \rangle;$$

$$S11 := \begin{bmatrix} -2.807 \\ -82.111 \\ 0 \end{bmatrix}$$

$$> S22 := \langle -86.769, -1.732, 13.092 \rangle;$$

$$S22 := \begin{bmatrix} -86.769 \\ -1.732 \\ 13.092 \end{bmatrix}$$

$$> S33 := \langle -78.809, 0, 0 \rangle;$$

$$S33 := \begin{bmatrix} -78.809 \\ 0 \\ 0 \end{bmatrix}$$

Matrices de Inercia del eslabón i con respecto a su centro de masa expresado en Si
Como toda la masa se considera concentrada en los centros de gravedad, las matrices de inercia serán

> $I11 := \langle (2422.309|0|0), (4.285|1587.979|0), (-0.002|-0.002|1474.055) \rangle;$

$$I11 := \begin{bmatrix} 2422.309 & 0 & 0 \\ 4.285 & 1587.979 & 0 \\ -0.002 & -0.002 & 1474.055 \end{bmatrix}$$

> $I22 := \langle (1059.248|0|0), (1.678|2089.822|0), (-99.320|43.420|1594.025) \rangle;$

$$I22 := \begin{bmatrix} 1059.248 & 0 & 0 \\ 1.678 & 2089.822 & 0 \\ -99.320 & 43.420 & 1594.025 \end{bmatrix}$$

> $I33 := \langle (282.163|0|0), (0|284.792|0), (0|0|262.959) \rangle;$

$$I33 := \begin{bmatrix} 282.163 & 0 & 0 \\ 0 & 284.792 & 0 \\ 0 & 0 & 262.959 \end{bmatrix}$$

Velocidades Angulares del sistema Si.

> $VelAng1i + Zo \cdot q1p;$

$$\begin{bmatrix} 0 \\ 0 \\ q1p \end{bmatrix}$$

> $Multiply(A01inv, \%);$

$$\begin{bmatrix} 0 \\ q1p \\ 0 \end{bmatrix}$$

> $VelAng11 := \%$

$$VelAng11 := \begin{bmatrix} 0 \\ q1p \\ 0 \end{bmatrix}$$

> $VelAng11 + Zo \cdot q2p;$

$$\begin{bmatrix} 0 \\ q1p \\ q2p \end{bmatrix}$$

> $Multiply(A12inv, \%);$

$$\begin{bmatrix} \sin(\theta_2) q1p \\ \cos(\theta_2) q1p \\ q2p \end{bmatrix}$$

> $VelAng22 := \%$

$$VelAng22 := \begin{bmatrix} \sin(\theta_2) q1p \\ \cos(\theta_2) q1p \\ q2p \end{bmatrix}$$

> $VelAng22 + Zo \cdot q3p;$

$$\begin{bmatrix} \sin(\theta_2) q1p \\ \cos(\theta_2) q1p \\ q2p + q3p \end{bmatrix}$$

> $Multiply(A23inv, \%);$

$$\begin{bmatrix} \cos(\theta_3) \sin(\theta_2) q1p + \sin(\theta_3) \cos(\theta_2) q1p \\ -\sin(\theta_3) \sin(\theta_2) q1p + \cos(\theta_3) \cos(\theta_2) q1p \\ q2p + q3p \end{bmatrix}$$

> $VelAng33 := \%;$

$$VelAng33 := \begin{bmatrix} \cos(\theta_3) \sin(\theta_2) q1p + \sin(\theta_3) \cos(\theta_2) q1p \\ -\sin(\theta_3) \sin(\theta_2) q1p + \cos(\theta_3) \cos(\theta_2) q1p \\ q2p + q3p \end{bmatrix}$$

Aceleraciones angulares del sistema Si

> $AceAngIni + Zo \cdot q1pp;$

$$\begin{bmatrix} 0 \\ 0 \\ q1pp \end{bmatrix}$$

> $Multiply(A01inv, \%);$

$$\begin{bmatrix} 0 \\ q1pp \\ 0 \end{bmatrix}$$

> $\% + CrossProduct(VelAngIni, Zo \cdot q1p);$

$$\begin{bmatrix} 0 \\ q1pp \\ 0 \end{bmatrix}$$

> $AceAng11 := \%;$

$$AceAng11 := \begin{bmatrix} 0 \\ q1pp \\ 0 \end{bmatrix}$$

> $AceAng11 + Zo \cdot q2pp;$

$$\begin{bmatrix} 0 \\ q1pp \\ q2pp \end{bmatrix}$$

> $Multiply(A12inv, \%);$

$$\begin{bmatrix} \sin(\theta_2) q1pp \\ \cos(\theta_2) q1pp \\ q2pp \end{bmatrix}$$

> % + CrossProduct(VelAng11, Zo· q2p);

$$\begin{bmatrix} \sin(\theta_2) q1pp + q1p q2p \\ \cos(\theta_2) q1pp \\ q2pp \end{bmatrix}$$

> AceAng22:= %;

$$AceAng22 := \begin{bmatrix} \sin(\theta_2) q1pp + q1p q2p \\ \cos(\theta_2) q1pp \\ q2pp \end{bmatrix}$$

> AceAng22 + Zo· q3pp;

$$\begin{bmatrix} \sin(\theta_2) q1pp + q1p q2p \\ \cos(\theta_2) q1pp \\ q2pp + q3pp \end{bmatrix}$$

> Multiply(A23inv,%);

$$\begin{bmatrix} \cos(\theta_3) (\sin(\theta_2) q1pp + q1p q2p) + \sin(\theta_3) \cos(\theta_2) q1pp \\ -\sin(\theta_3) (\sin(\theta_2) q1pp + q1p q2p) + \cos(\theta_3) \cos(\theta_2) q1pp \\ q2pp + q3pp \end{bmatrix}$$

> % + CrossProduct(VelAng22, Zo· q3p);

$$\begin{bmatrix} [\cos(\theta_3) (\sin(\theta_2) q1pp + q1p q2p) + \sin(\theta_3) \cos(\theta_2) q1pp + \cos(\theta_2) q1p q3p], \\ [-\sin(\theta_3) (\sin(\theta_2) q1pp + q1p q2p) + \cos(\theta_3) \cos(\theta_2) q1pp \\ - \sin(\theta_2) q1p q3p], \\ [q2pp + q3pp] \end{bmatrix}$$

> AceAng33:= %;

$$AceAng33 := \begin{bmatrix} [\cos(\theta_3) (\sin(\theta_2) q1pp + q1p q2p) + \sin(\theta_3) \cos(\theta_2) q1pp \\ + \cos(\theta_2) q1p q3p], \\ [-\sin(\theta_3) (\sin(\theta_2) q1pp + q1p q2p) + \cos(\theta_3) \cos(\theta_2) q1pp \\ - \sin(\theta_2) q1p q3p], \\ [q2pp + q3pp] \end{bmatrix}$$

Aceleraciones lineales del sistema i

> AceLin11 := CrossProduct(AceAng11, P11) + CrossProduct(VelAng11, CrossProduct(VelAng11, P11)) + Multiply(A01inv, AceLin1ni);

$$AceLin11 := \begin{bmatrix} 0 \\ gz \\ 0 \end{bmatrix}$$

> AceLin22 := CrossProduct(AceAng22, P22) + CrossProduct(VelAng22, CrossProduct(VelAng22, P22)) + Multiply(A12inv, AceLin11);

$$AceLin22 := \begin{bmatrix} -200 \cos(\theta_2)^2 q1p^2 - 200 q2p^2 + \sin(\theta_2) gz \\ 200 q2pp + 200 \sin(\theta_2) q1p^2 \cos(\theta_2) + \cos(\theta_2) gz \\ -200 \cos(\theta_2) q1pp + 200 \sin(\theta_2) q1p q2p \end{bmatrix}$$

Licencia Creative Commons:

Atribución - No Comercial - Compartir Igual 3.0 Venezuela
(CC BY-NC-SA 3.0 VE)

$$AceLin33 := CrossProduct(AceAng33, P33) + CrossProduct(VelAng33, CrossProduct(VelAng33, P33)) + Multiply(A23inv, AceLin22);$$

$$AceLin33 := \left[\left[\begin{aligned} &(-\sin(\theta_3) \sin(\theta_2) q1p \\ &+ \cos(\theta_3) \cos(\theta_2) q1p) (130 \sin(\theta_3) \sin(\theta_2) q1p \\ &- 130 \cos(\theta_3) \cos(\theta_2) q1p) - (q2p + q3p) (130 q2p + 130 q3p) + \cos(\theta_3) (\\ &- 200 \cos(\theta_2)^2 q1p^2 - 200 q2p^2 + \sin(\theta_2) gz) + \sin(\theta_3) (200 q2pp \\ &+ 200 \sin(\theta_2) q1p^2 \cos(\theta_2) + \cos(\theta_2) gz) \end{aligned} \right], \right. \\ \left. \left[\begin{aligned} &130 q2pp + 130 q3pp - (\cos(\theta_3) \sin(\theta_2) q1p \\ &+ \sin(\theta_3) \cos(\theta_2) q1p) (130 \sin(\theta_3) \sin(\theta_2) q1p \\ &- 130 \cos(\theta_3) \cos(\theta_2) q1p) - \sin(\theta_3) (-200 \cos(\theta_2)^2 q1p^2 - 200 q2p^2 \\ &+ \sin(\theta_2) gz) + \cos(\theta_3) (200 q2pp + 200 \sin(\theta_2) q1p^2 \cos(\theta_2) \\ &+ \cos(\theta_2) gz) \end{aligned} \right], \right. \\ \left. \left[\begin{aligned} &130 \sin(\theta_3) (\sin(\theta_2) q1pp + q1p q2p) - 130 \cos(\theta_3) \cos(\theta_2) q1pp \\ &+ 130 \sin(\theta_2) q1p q3p + (\cos(\theta_3) \sin(\theta_2) q1p \\ &+ \sin(\theta_3) \cos(\theta_2) q1p) (130 q2p + 130 q3p) - 200 \cos(\theta_2) q1pp \\ &+ 200 \sin(\theta_2) q1p q2p \end{aligned} \right] \right]$$

Aceleraciones lineales del centro de gravedad del eslabon i

$$A11 := CrossProduct(AceAng11, S11) + CrossProduct(VelAng11, CrossProduct(VelAng11, S11)) + AceLin1;$$

$$A11 := \begin{bmatrix} 2.807 q1p^2 \\ gz \\ 2.807 q1pp \end{bmatrix}$$

$$A22 := CrossProduct(AceAng22, S22) + CrossProduct(VelAng22, CrossProduct(VelAng22, S22)) + AceLin2;$$

$$A22 := \left[\left[\begin{aligned} &13.092 \cos(\theta_2) q1pp + 1.732 q2pp + \cos(\theta_2) q1p (-1.732 \sin(\theta_2) q1p \\ &+ 86.769 \cos(\theta_2) q1p) - q2p (-86.769 q2p - 13.092 \sin(\theta_2) q1p) \\ &- 200 \cos(\theta_2)^2 q1p^2 - 200 q2p^2 + \sin(\theta_2) gz \end{aligned} \right], \right. \\ \left. \left[\begin{aligned} &113.231 q2pp - 13.092 \sin(\theta_2) q1pp - 13.092 q1p q2p \\ &+ q2p (13.092 \cos(\theta_2) q1p + 1.732 q2p) - \sin(\theta_2) q1p (-1.732 \sin(\theta_2) q1p \\ &+ 86.769 \cos(\theta_2) q1p) + 200 \sin(\theta_2) q1p^2 \cos(\theta_2) + \cos(\theta_2) gz \end{aligned} \right], \right. \\ \left. \left[\begin{aligned} &-1.732 \sin(\theta_2) q1pp - 1.732 q1p q2p - 113.231 \cos(\theta_2) q1pp + \sin(\theta_2) q1p (\\ &-86.769 q2p - 13.092 \sin(\theta_2) q1p) - \cos(\theta_2) q1p (13.092 \cos(\theta_2) q1p \\ &+ 1.732 q2p) + 200 \sin(\theta_2) q1p q2p \end{aligned} \right] \right]$$

$$A33 := CrossProduct(AceAng33, S33) + CrossProduct(VelAng33, CrossProduct(VelAng33, S33)) + AceLin3;$$

$$\begin{aligned}
A_{33} := & \left[\left(-\sin(\theta_3) \sin(\theta_2) q_1 p + \cos(\theta_3) \cos(\theta_2) q_1 p \right) \left(-78.809 \sin(\theta_3) \sin(\theta_2) q_1 p + 78.809 \cos(\theta_3) \cos(\theta_2) q_1 p \right) - (q_2 p + q_3 p) \left(-78.809 q_2 p - 78.809 q_3 p \right) + \left(-\sin(\theta_3) \sin(\theta_2) q_1 p + \cos(\theta_3) \cos(\theta_2) q_1 p \right) \left(130 \sin(\theta_3) \sin(\theta_2) q_1 p - 130 \cos(\theta_3) \cos(\theta_2) q_1 p \right) - (q_2 p + q_3 p) \left(130 q_2 p + 130 q_3 p \right) + \cos(\theta_3) \left(-200 \cos(\theta_2)^2 q_1 p^2 - 200 q_2 p^2 + \sin(\theta_2) g z \right) + \sin(\theta_3) \left(200 q_2 p p + 200 \sin(\theta_2) q_1 p^2 \cos(\theta_2) + \cos(\theta_2) g z \right) \right], \\
& \left[51.191 q_2 p p + 51.191 q_3 p p - \left(\cos(\theta_3) \sin(\theta_2) q_1 p + \sin(\theta_3) \cos(\theta_2) q_1 p \right) \left(-78.809 \sin(\theta_3) \sin(\theta_2) q_1 p + 78.809 \cos(\theta_3) \cos(\theta_2) q_1 p \right) - \left(\cos(\theta_3) \sin(\theta_2) q_1 p + \sin(\theta_3) \cos(\theta_2) q_1 p \right) \left(130 \sin(\theta_3) \sin(\theta_2) q_1 p - 130 \cos(\theta_3) \cos(\theta_2) q_1 p \right) - \sin(\theta_3) \left(-200 \cos(\theta_2)^2 q_1 p^2 - 200 q_2 p^2 + \sin(\theta_2) g z \right) + \cos(\theta_3) \left(200 q_2 p p + 200 \sin(\theta_2) q_1 p^2 \cos(\theta_2) + \cos(\theta_2) g z \right) \right], \\
& \left[51.191 \sin(\theta_3) \left(\sin(\theta_2) q_1 p p + q_1 p q_2 p \right) - 51.191 \cos(\theta_3) \cos(\theta_2) q_1 p p + 51.191 \sin(\theta_2) q_1 p q_3 p + \left(\cos(\theta_3) \sin(\theta_2) q_1 p + \sin(\theta_3) \cos(\theta_2) q_1 p \right) \left(-78.809 q_2 p - 78.809 q_3 p \right) + \left(\cos(\theta_3) \sin(\theta_2) q_1 p + \sin(\theta_3) \cos(\theta_2) q_1 p \right) \left(130 q_2 p + 130 q_3 p \right) - 200 \cos(\theta_2) q_1 p p + 200 \sin(\theta_2) q_1 p q_2 p \right]
\end{aligned}$$

Fuerza ejercida sobre el eslabón i

Primero definiremos las masas de cada eslabón:

Masa para el eslabón 1, compuesto por la base redonda, el perfil, dos extensiones y el motor. Kg

$$m_1 := 0.583;$$

$$m_1 := 0.583$$

Masa para el eslabon 2, compuesto por el perfil, dos extensiones y el motor. Kg

$$m_2 := 0.473;$$

$$m_2 := 0.473$$

Masa para el eslabón 3, compuesto por el perfil y el motor. Kg

$$m_3 := 0.153;$$

$$m_3 := 0.153$$

Una vez definidas las masas para los eslabones se calcula la fuerza ejercida sobre los mismos.

Las mismas se calculan de manera recursiva desde el último eslabón hasta el primero.

El primero que se calcula es el del eslabón 2, ya que sobre el eslabón 3 se asume como condición inicial

que no estuviera actuando ninguna carga o fuerza.

$$f_{33} := m_3 \cdot A_{33};$$

Licencia Creative Commons:

Atribución - No Comercial - Compartir Igual 3.0 Venezuela
(CC BY-NC-SA 3.0 VE)

$$\begin{aligned}
f_{33} := & \left[\left[0.153 (-\sin(\theta_3) \sin(\theta_2) q_1 p + \cos(\theta_3) \cos(\theta_2) q_1 p) (\right. \right. \\
& - 78.809 \sin(\theta_3) \sin(\theta_2) q_1 p + 78.809 \cos(\theta_3) \cos(\theta_2) q_1 p) - 0.153 (q_2 p \\
& + q_3 p) (-78.809 q_2 p - 78.809 q_3 p) + 0.153 (-\sin(\theta_3) \sin(\theta_2) q_1 p \\
& + \cos(\theta_3) \cos(\theta_2) q_1 p) (130 \sin(\theta_3) \sin(\theta_2) q_1 p \\
& - 130 \cos(\theta_3) \cos(\theta_2) q_1 p) - 0.153 (q_2 p + q_3 p) (130 q_2 p + 130 q_3 p) \\
& + 0.153 \cos(\theta_3) (-200 \cos(\theta_2)^2 q_1 p^2 - 200 q_2 p^2 + \sin(\theta_2) g_z) \\
& \left. \left. + 0.153 \sin(\theta_3) (200 q_2 p p + 200 \sin(\theta_2) q_1 p^2 \cos(\theta_2) + \cos(\theta_2) g_z) \right], \right. \\
& \left[7.832223 q_2 p p + 7.832223 q_3 p p - 0.153 (\cos(\theta_3) \sin(\theta_2) q_1 p \right. \\
& + \sin(\theta_3) \cos(\theta_2) q_1 p) (-78.809 \sin(\theta_3) \sin(\theta_2) q_1 p \\
& + 78.809 \cos(\theta_3) \cos(\theta_2) q_1 p) - 0.153 (\cos(\theta_3) \sin(\theta_2) q_1 p \\
& + \sin(\theta_3) \cos(\theta_2) q_1 p) (130 \sin(\theta_3) \sin(\theta_2) q_1 p \\
& - 130 \cos(\theta_3) \cos(\theta_2) q_1 p) - 0.153 \sin(\theta_3) (-200 \cos(\theta_2)^2 q_1 p^2 - 200 q_2 p^2 \\
& + \sin(\theta_2) g_z) + 0.153 \cos(\theta_3) (200 q_2 p p + 200 \sin(\theta_2) q_1 p^2 \cos(\theta_2) \\
& \left. \left. + \cos(\theta_2) g_z) \right], \right. \\
& \left[7.832223 \sin(\theta_3) (\sin(\theta_2) q_1 p p + q_1 p q_2 p) - 7.832223 \cos(\theta_3) \cos(\theta_2) q_1 p p \right. \\
& + 7.832223 \sin(\theta_2) q_1 p q_3 p + 0.153 (\cos(\theta_3) \sin(\theta_2) q_1 p \\
& + \sin(\theta_3) \cos(\theta_2) q_1 p) (-78.809 q_2 p - 78.809 q_3 p) \\
& + 0.153 (\cos(\theta_3) \sin(\theta_2) q_1 p + \sin(\theta_3) \cos(\theta_2) q_1 p) (130 q_2 p + 130 q_3 p) \\
& \left. \left. - 30.600 \cos(\theta_2) q_1 p p + 30.600 \sin(\theta_2) q_1 p q_2 p \right] \right]
\end{aligned}$$

$$f_{22} := \text{Multiply}(A_{23}, f_{33}) + m_2 \cdot A_{22};$$

WWW.BDIGITAL.ULA.VE

$$\begin{aligned}
 f_{22} := & \left[\cos(\theta_3) (0.153 (-\sin(\theta_3) \sin(\theta_2) q_1 p + \cos(\theta_3) \cos(\theta_2) q_1 p) (\right. \\
 & - 78.809 \sin(\theta_3) \sin(\theta_2) q_1 p + 78.809 \cos(\theta_3) \cos(\theta_2) q_1 p) - 0.153 (q_2 p \\
 & + q_3 p) (-78.809 q_2 p - 78.809 q_3 p) + 0.153 (-\sin(\theta_3) \sin(\theta_2) q_1 p \\
 & + \cos(\theta_3) \cos(\theta_2) q_1 p) (130 \sin(\theta_3) \sin(\theta_2) q_1 p \\
 & - 130 \cos(\theta_3) \cos(\theta_2) q_1 p) - 0.153 (q_2 p + q_3 p) (130 q_2 p + 130 q_3 p) \\
 & + 0.153 \cos(\theta_3) (-200 \cos(\theta_2)^2 q_1 p^2 - 200 q_2 p^2 + \sin(\theta_2) g z) \\
 & + 0.153 \sin(\theta_3) (200 q_2 p p + 200 \sin(\theta_2) q_1 p^2 \cos(\theta_2) + \cos(\theta_2) g z) \\
 & - \sin(\theta_3) (7.832223 q_2 p p + 7.832223 q_3 p p - 0.153 (\cos(\theta_3) \sin(\theta_2) q_1 p \\
 & + \sin(\theta_3) \cos(\theta_2) q_1 p) (-78.809 \sin(\theta_3) \sin(\theta_2) q_1 p \\
 & + 78.809 \cos(\theta_3) \cos(\theta_2) q_1 p) - 0.153 (\cos(\theta_3) \sin(\theta_2) q_1 p \\
 & + \sin(\theta_3) \cos(\theta_2) q_1 p) (130 \sin(\theta_3) \sin(\theta_2) q_1 p \\
 & - 130 \cos(\theta_3) \cos(\theta_2) q_1 p) - 0.153 \sin(\theta_3) (-200 \cos(\theta_2)^2 q_1 p^2 - 200 q_2 p^2 \\
 & + \sin(\theta_2) g z) + 0.153 \cos(\theta_3) (200 q_2 p p + 200 \sin(\theta_2) q_1 p^2 \cos(\theta_2) \\
 & + \cos(\theta_2) g z) \left. \right) + 6.192516 \cos(\theta_2) q_1 p p + 0.819236 q_2 p p \\
 & + 0.473 \cos(\theta_2) q_1 p (-1.732 \sin(\theta_2) q_1 p + 86.769 \cos(\theta_2) q_1 p) \\
 & - 0.473 q_2 p (-86.769 q_2 p - 13.092 \sin(\theta_2) q_1 p) - 94.600 \cos(\theta_2)^2 q_1 p^2 \\
 & - 94.600 q_2 p^2 + 0.473 \sin(\theta_2) g z \left. \right], \\
 & \left[\sin(\theta_3) (0.153 (-\sin(\theta_3) \sin(\theta_2) q_1 p + \cos(\theta_3) \cos(\theta_2) q_1 p) (\right. \\
 & - 78.809 \sin(\theta_3) \sin(\theta_2) q_1 p + 78.809 \cos(\theta_3) \cos(\theta_2) q_1 p) - 0.153 (q_2 p \\
 & + q_3 p) (-78.809 q_2 p - 78.809 q_3 p) + 0.153 (-\sin(\theta_3) \sin(\theta_2) q_1 p \\
 & + \cos(\theta_3) \cos(\theta_2) q_1 p) (130 \sin(\theta_3) \sin(\theta_2) q_1 p \\
 & - 130 \cos(\theta_3) \cos(\theta_2) q_1 p) - 0.153 (q_2 p + q_3 p) (130 q_2 p + 130 q_3 p) \\
 & + 0.153 \cos(\theta_3) (-200 \cos(\theta_2)^2 q_1 p^2 - 200 q_2 p^2 + \sin(\theta_2) g z) \\
 & + 0.153 \sin(\theta_3) (200 q_2 p p + 200 \sin(\theta_2) q_1 p^2 \cos(\theta_2) + \cos(\theta_2) g z) \\
 & + \cos(\theta_3) (7.832223 q_2 p p + 7.832223 q_3 p p - 0.153 (\cos(\theta_3) \sin(\theta_2) q_1 p \\
 & + \sin(\theta_3) \cos(\theta_2) q_1 p) (-78.809 \sin(\theta_3) \sin(\theta_2) q_1 p \\
 & + 78.809 \cos(\theta_3) \cos(\theta_2) q_1 p) - 0.153 (\cos(\theta_3) \sin(\theta_2) q_1 p \\
 & + \sin(\theta_3) \cos(\theta_2) q_1 p) (130 \sin(\theta_3) \sin(\theta_2) q_1 p \\
 & - 130 \cos(\theta_3) \cos(\theta_2) q_1 p) - 0.153 \sin(\theta_3) (-200 \cos(\theta_2)^2 q_1 p^2 - 200 q_2 p^2 \\
 & + \sin(\theta_2) g z) + 0.153 \cos(\theta_3) (200 q_2 p p + 200 \sin(\theta_2) q_1 p^2 \cos(\theta_2) \\
 & + \cos(\theta_2) g z) \left. \right) + 53.558263 q_2 p p - 6.192516 \sin(\theta_2) q_1 p p \\
 & - 6.192516 q_1 p q_2 p + 0.473 q_2 p (13.092 \cos(\theta_2) q_1 p + 1.732 q_2 p) \\
 & - 0.473 \sin(\theta_2) q_1 p (-1.732 \sin(\theta_2) q_1 p + 86.769 \cos(\theta_2) q_1 p) \\
 & + 94.600 \sin(\theta_2) q_1 p^2 \cos(\theta_2) + 0.473 \cos(\theta_2) g z \left. \right], \\
 & \left[7.832223 \sin(\theta_3) (\sin(\theta_2) q_1 p p + q_1 p q_2 p) - 7.832223 \cos(\theta_3) \cos(\theta_2) q_1 p, \right. \\
 & + 7.832223 \sin(\theta_2) q_1 p q_3 p + 0.153 (\cos(\theta_3) \sin(\theta_2) q_1 p \\
 & + \sin(\theta_3) \cos(\theta_2) q_1 p) (-78.809 q_2 p - 78.809 q_3 p) \\
 & + 0.153 (\cos(\theta_3) \sin(\theta_2) q_1 p + \sin(\theta_3) \cos(\theta_2) q_1 p) (130 q_2 p + 130 q_3 p) \\
 & - 84.158263 \cos(\theta_2) q_1 p p + 125.200 \sin(\theta_2) q_1 p q_2 p - 0.819236 \sin(\theta_2) q_1 p p \\
 & - (819236 q_1 p q_2 p) - 0.473 \sin(\theta_2) q_1 p (-86.769 q_2 p - 13.092 \sin(\theta_2) q_1 p) \\
 & \left. - 0.473 \cos(\theta_2) q_1 p (13.092 \cos(\theta_2) q_1 p + 1.732 q_2 p) \right]
 \end{aligned}$$

```
> f11 := Multiply(A12, f22) + m1 · A11;
```

WWW.BDIGITAL.ULA.VE

Licencia Creative Commons:
Atribución - No Comercial - Compartir Igual 3.0 Venezuela
(CC BY-NC-SA 3.0 VE)

Par ejercicio sobre el eslabón i

Al igual que en las fuerzas ejercidas sobre los eslabones, los pares se calculan de manera recursiva, comenzando por el último eslabón.

$$\triangleright A01 := \langle (1 | 0 | 0), (0 | 1 | 0), (0 | 0 | 1) \rangle;$$

$$A01 := \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\triangleright n44 := \langle 0, 0, 0 \rangle;$$

$$n44 := \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

$$\triangleright n33 := \text{CrossProduct}(P33 + S33, m3 \cdot A33) + \text{Multiply}(I33, \text{AceAng33}) \\ + \text{CrossProduct}(\text{VelAng33}, \text{Multiply}(I33, \text{VelAng33}));$$

$$\begin{aligned} n33 := & [[282.163 \cos(\theta_3) (\sin(\theta_2) q1pp + q1p q2p) \\ & + 282.163 \sin(\theta_3) \cos(\theta_2) q1pp + 282.163 \cos(\theta_2) q1p q3p + (\\ & -\sin(\theta_3) \sin(\theta_2) q1p + \cos(\theta_3) \cos(\theta_2) q1p) (262.959 q2p + 262.959 q3p) \\ & - (q2p + q3p) (-284.792 \sin(\theta_3) \sin(\theta_2) q1p \\ & + 284.792 \cos(\theta_3) \cos(\theta_2) q1p)], \\ & [-685.7313276 \sin(\theta_3) (\sin(\theta_2) q1pp + q1p q2p) \\ & + 685.7313276 \cos(\theta_3) \cos(\theta_2) q1pp - 685.7313276 \sin(\theta_2) q1p q3p \\ & - 7.832223000 (\cos(\theta_3) \sin(\theta_2) q1p + \sin(\theta_3) \cos(\theta_2) q1p) (-78.809 q2p \\ & - 78.809 q3p) - 7.832223000 (\cos(\theta_3) \sin(\theta_2) q1p \\ & + \sin(\theta_3) \cos(\theta_2) q1p) (130 q2p + 130 q3p) + 1566.444600 \cos(\theta_2) q1pp \\ & - 1566.444600 \sin(\theta_2) q1p q2p + (q2p + q3p) (282.163 \cos(\theta_3) \sin(\theta_2) q1p \\ & + 282.163 \sin(\theta_3) \cos(\theta_2) q1p) - (\cos(\theta_3) \sin(\theta_2) q1p \\ & + \sin(\theta_3) \cos(\theta_2) q1p) (262.959 q2p + 262.959 q3p)], \\ & [663.8983276 q2pp + 663.8983276 q3pp - 7.832223000 (\cos(\theta_3) \sin(\theta_2) q1p \\ & + \sin(\theta_3) \cos(\theta_2) q1p) (-78.809 \sin(\theta_3) \sin(\theta_2) q1p \\ & + 78.809 \cos(\theta_3) \cos(\theta_2) q1p) - 7.832223000 (\cos(\theta_3) \sin(\theta_2) q1p \\ & + \sin(\theta_3) \cos(\theta_2) q1p) (130 \sin(\theta_3) \sin(\theta_2) q1p \\ & - 130 \cos(\theta_3) \cos(\theta_2) q1p) - 7.832223000 \sin(\theta_3) (-200 \cos(\theta_2)^2 q1p^2 \\ & - 200 q2p^2 + \sin(\theta_2) gz) + 7.832223000 \cos(\theta_3) (200 q2pp \\ & + 200 \sin(\theta_2) q1p^2 \cos(\theta_2) + \cos(\theta_2) gz) + (\cos(\theta_3) \sin(\theta_2) q1p \\ & + \sin(\theta_3) \cos(\theta_2) q1p) (-284.792 \sin(\theta_3) \sin(\theta_2) q1p \\ & + 284.792 \cos(\theta_3) \cos(\theta_2) q1p) - (-\sin(\theta_3) \sin(\theta_2) q1p \\ & + \cos(\theta_3) \cos(\theta_2) q1p) (282.163 \cos(\theta_3) \sin(\theta_2) q1p \\ & + 282.163 \sin(\theta_3) \cos(\theta_2) q1p)]] \end{aligned}$$

> $n22 := \text{Multiply}(A23, (n33 + \text{CrossProduct}(\text{Multiply}(A23\text{inv}, P22), f33)))$
 + $\text{CrossProduct}(P22 + S22, m2 \cdot A22) + \text{Multiply}(I22, \text{VelAng22})$
 + $\text{CrossProduct}(\text{VelAng22}, \text{Multiply}(I22, \text{VelAng22}));$

> $n11 := \text{Multiply}(A12, (n22 + \text{CrossProduct}(\text{Multiply}(A12\text{inv}, P11), f22)))$
 + $\text{CrossProduct}(P11 + S11, m1 \cdot A11) + \text{Multiply}(I11, \text{VelAng11})$
 + $\text{CrossProduct}(\text{VelAng11}, \text{Multiply}(I11, \text{VelAng11}));$

> **Fuerza o par aplicado a la articulación i**

Finalmente se calcula la fuerza o par aplicado a la articulación, para así poder calcular el motor a utilizar.

> $T1 := \text{Multiply}(\text{Transpose}(n11), \text{Multiply}(A01\text{inv}, Zo));$

WWW.BDIGITAL.ULA.VE

$$\begin{aligned}
T1 := & \sin(\theta_2) \left(\cos(\theta_3) \left(282.163 \cos(\theta_3) \left(\sin(\theta_2) q1pp + q1p q2p \right) \right. \right. \\
& + 282.163 \sin(\theta_3) \cos(\theta_2) q1pp + 282.163 \cos(\theta_2) q1p q3p + \left. \left. \right. \right. \\
& - \sin(\theta_3) \sin(\theta_2) q1p + \cos(\theta_3) \cos(\theta_2) q1p \left. \left. \left(262.959 q2p + 262.959 q3p \right) \right. \right. \\
& - \left. \left. \left(q2p + q3p \right) \left(-284.792 \sin(\theta_3) \sin(\theta_2) q1p \right) \right. \right. \\
& + 284.792 \cos(\theta_3) \cos(\theta_2) q1p \left. \left. \right. \right. \\
& - 200 \sin(\theta_3) \left(7.832223 \sin(\theta_3) \left(\sin(\theta_2) q1pp + q1p q2p \right) \right. \\
& - 7.832223 \cos(\theta_3) \cos(\theta_2) q1pp + 7.832223 \sin(\theta_2) q1p q3p \\
& + 0.153 \left(\cos(\theta_3) \sin(\theta_2) q1p + \sin(\theta_3) \cos(\theta_2) q1p \right) \left(-78.809 q2p \right. \\
& - 78.809 q3p \left. \right) + 0.153 \left(\cos(\theta_3) \sin(\theta_2) q1p + \sin(\theta_3) \cos(\theta_2) q1p \right) \left(130 q2p \right. \\
& + 130 q3p \left. \right) - 30.600 \cos(\theta_2) q1pp + 30.600 \sin(\theta_2) q1p q2p \left. \right) - \sin(\theta_3) \left(\right. \\
& - 685.7313276 \sin(\theta_3) \left(\sin(\theta_2) q1pp + q1p q2p \right) \\
& + 685.7313276 \cos(\theta_3) \cos(\theta_2) q1pp - 685.7313276 \sin(\theta_2) q1p q3p \\
& - 7.832223000 \left(\cos(\theta_3) \sin(\theta_2) q1p + \sin(\theta_3) \cos(\theta_2) q1p \right) \left(-78.809 q2p \right. \\
& - 78.809 q3p \left. \right) - 7.832223000 \left(\cos(\theta_3) \sin(\theta_2) q1p \right. \\
& + \sin(\theta_3) \cos(\theta_2) q1p \left. \right) \left(130 q2p + 130 q3p \right) + 1566.444600 \cos(\theta_2) q1pp \\
& - 1566.444600 \sin(\theta_2) q1p q2p + \left(q2p + q3p \right) \left(282.163 \cos(\theta_3) \sin(\theta_2) q1p \right. \\
& + 282.163 \sin(\theta_3) \cos(\theta_2) q1p \left. \right) - \left(\cos(\theta_3) \sin(\theta_2) q1p \right. \\
& + \sin(\theta_3) \cos(\theta_2) q1p \left. \right) \left(262.959 q2p + 262.959 q3p \right) \\
& - 200 \cos(\theta_3) \left(7.832223 \sin(\theta_3) \left(\sin(\theta_2) q1pp + q1p q2p \right) \right. \\
& - 7.832223 \cos(\theta_3) \cos(\theta_2) q1pp + 7.832223 \sin(\theta_2) q1p q3p \\
& + 0.153 \left(\cos(\theta_3) \sin(\theta_2) q1p + \sin(\theta_3) \cos(\theta_2) q1p \right) \left(-78.809 q2p \right. \\
& - 78.809 q3p \left. \right) + 0.153 \left(\cos(\theta_3) \sin(\theta_2) q1p + \sin(\theta_3) \cos(\theta_2) q1p \right) \left(130 q2p \right. \\
& + 130 q3p \left. \right) - 30.600 \cos(\theta_2) q1pp + 30.600 \sin(\theta_2) q1p q2p \left. \right) \\
& + 82.49133622 \sin(\theta_2) q1pp + 82.49133622 q1p q2p \\
& + 92.76291152 \cos(\theta_2) q1pp - 0.8192360000 \sin(\theta_2) q1p \left(-86.769 q2p \right. \\
& - 13.092 \sin(\theta_2) q1p \left. \right) + 0.8192360000 \cos(\theta_2) q1p \left(13.092 \cos(\theta_2) q1p \right. \\
& + 1.732 q2p \left. \right) - 163.8472000 \sin(\theta_2) q1p q2p - 701.1847792 q2pp \\
& - 6.192516000 q2p \left(13.092 \cos(\theta_2) q1p + 1.732 q2p \right) \\
& + 6.192516000 \sin(\theta_2) q1p \left(-1.732 \sin(\theta_2) q1p + 86.769 \cos(\theta_2) q1p \right) \\
& - 1238.503200 \sin(\theta_2) q1p^2 \cos(\theta_2) - 6.192516000 \cos(\theta_2) gz \\
& + 1059.248 \sin(\theta_2) q1p + \cos(\theta_2) q1p \left(-99.320 \sin(\theta_2) q1p \right. \\
& + 43.420 \cos(\theta_2) q1p + 1594.025 q2p \left. \right) - q2p \left(1.678 \sin(\theta_2) q1p \right. \\
& + 2089.822 \cos(\theta_2) q1p \left. \right) + 230 \cos(\theta_2) \left(7.832223 \sin(\theta_3) \left(\sin(\theta_2) q1pp \right. \right. \\
& + \left. \left. q1p q2p \right) - 7.832223 \cos(\theta_3) \cos(\theta_2) q1pp + 7.832223 \sin(\theta_2) q1p q3p \right. \\
& + 0.153 \left(\cos(\theta_3) \sin(\theta_2) q1p + \sin(\theta_3) \cos(\theta_2) q1p \right) \left(-78.809 q2p \right. \\
& - 78.809 q3p \left. \right) + 0.153 \left(\cos(\theta_3) \sin(\theta_2) q1p + \sin(\theta_3) \cos(\theta_2) q1p \right) \left(130 q2p \right. \\
& + 130 q3p \left. \right) - 84.158263 \cos(\theta_2) q1pp + 125.200 \sin(\theta_2) q1p q2p \\
& - 0.819236 \sin(\theta_2) q1pp - 0.819236 q1p q2p + 0.473 \sin(\theta_2) q1p \left(\right. \\
& - 86.769 q2p - 13.092 \sin(\theta_2) q1p \left. \right) - 0.473 \cos(\theta_2) q1p \left(13.092 \cos(\theta_2) q1p \right. \\
& + 1.732 q2p \left. \right) \left. \right) + \cos(\theta_2) \left(\sin(\theta_3) \left(282.163 \cos(\theta_3) \left(\sin(\theta_2) q1pp \right. \right. \right. \\
& + \left. \left. q1p q2p \right) + 282.163 \sin(\theta_3) \cos(\theta_2) q1pp + 282.163 \cos(\theta_2) q1p q3p + \left. \left. \right. \right. \\
& - \left. \left. \sin(\theta_3) \sin(\theta_2) q1p + \cos(\theta_3) \cos(\theta_2) q1p \right) \left(262.959 q2p + 262.959 q3p \right) \right. \\
& - \left. \left. \left(q2p + q3p \right) \left(-284.792 \sin(\theta_3) \sin(\theta_2) q1p \right) \right. \right. \\
& + 284.792 \cos(\theta_3) \cos(\theta_2) q1p \left. \left. \right. \right)
\end{aligned}$$

```

>
> T2 := Multiply(Transpose(n22), Multiply(A12inv, Zo));
      T2 := (cos(θ3) sin(θ2) q1p + sin(θ3) cos(θ2) q1p) (
      -284.792 sin(θ3) sin(θ2) q1p + 284.792 cos(θ3) cos(θ2) q1p) - (
      -sin(θ3) sin(θ2) q1p + cos(θ3) cos(θ2) q1p) (282.163 cos(θ3) sin(θ2) q1p
      + 282.163 sin(θ3) cos(θ2) q1p) + 6729.772923 q2pp + 1594.025 q2p
      + 663.8983276 q3pp - 53.55826300 sin(θ2) q1p (-1.732 sin(θ2) q1p
      + 86.769 cos(θ2) q1p) - 701.1847792 sin(θ2) q1pp
      + 10.72543771 cos(θ2) q1pp + sin(θ2) q1p (1.678 sin(θ2) q1p
      + 2089.822 cos(θ2) q1p) + 7.832223000 cos(θ3) (200 q2pp
      + 200 sin(θ2) q1p2 cos(θ2) + cos(θ2) gz) - 7.832223000 sin(θ3) (
      -200 cos(θ2)2 q1p2 - 200 q2p2 + sin(θ2) gz) + 0.8192360000 cos(θ2) q1p (
      -1.732 sin(θ2) q1p + 86.769 cos(θ2) q1p) + 9652.40460 sin(θ2) q1p2 cos(θ2)
      - 0.8192360000 q2p (-86.769 q2p - 13.092 sin(θ2) q1p)
      + 53.55826300 q2p (13.092 cos(θ2) q1p + 1.732 q2p) - 99.320 sin(θ2) q1p
      + 43.420 cos(θ2) q1p - 7.832223000 (cos(θ3) sin(θ2) q1p
      + sin(θ3) cos(θ2) q1p) (130 sin(θ3) sin(θ2) q1p
      - 130 cos(θ3) cos(θ2) q1p) - 163.8472000 q2p2 + 200 sin(θ3) (0.153 (
      -sin(θ3) sin(θ2) q1p + cos(θ3) cos(θ2) q1p) (-78.809 sin(θ3) sin(θ2) q1p
      + 78.809 cos(θ3) cos(θ2) q1p) - 0.153 (q2p + q3p) (-78.809 q2p
      - 78.809 q3p) + 0.153 (-sin(θ3) sin(θ2) q1p
      + cos(θ3) cos(θ2) q1p) (130 sin(θ3) sin(θ2) q1p
      - 130 cos(θ3) cos(θ2) q1p) - 0.153 (q2p + q3p) (130 q2p + 130 q3p)
      + 0.153 cos(θ3) (-200 cos(θ2)2 q1p2 - 200 q2p2 + sin(θ2) gz)
      + 0.153 sin(θ3) (200 q2pp + 200 sin(θ2) q1p2 cos(θ2) + cos(θ2) gz))
      + 200 cos(θ3) (7.832223 q2pp + 7.832223 q3pp - 0.153 (cos(θ3) sin(θ2) q1p
      + sin(θ3) cos(θ2) q1p) (-78.809 sin(θ3) sin(θ2) q1p
      + 78.809 cos(θ3) cos(θ2) q1p) - 0.153 (cos(θ3) sin(θ2) q1p
      + sin(θ3) cos(θ2) q1p) (130 sin(θ3) sin(θ2) q1p
      - 130 cos(θ3) cos(θ2) q1p) - 0.153 sin(θ3) (-200 cos(θ2)2 q1p2 - 200 q2p2
      + sin(θ2) gz) + 0.153 cos(θ3) (200 q2pp + 200 sin(θ2) q1p2 cos(θ2)
      + cos(θ2) gz)) - 701.1847792 q1p q2p - 163.8472000 cos(θ2)2 q1p2
      + 0.8192360000 sin(θ2) gz + 53.55826300 cos(θ2) gz
      - 7.832223000 (cos(θ3) sin(θ2) q1p + sin(θ3) cos(θ2) q1p) (
      -78.809 sin(θ3) sin(θ2) q1p + 78.809 cos(θ3) cos(θ2) q1p)

```

```

>
> T3 := Multiply(Transpose(n33), Multiply(A23inv, Zo));

```

$$\begin{aligned}
T3 := & 663.8983276 q2pp + 663.8983276 q3pp - 7.832223000 (\cos(\theta3) \sin(\theta2) q1p) \\
& + \sin(\theta3) \cos(\theta2) q1p) (-78.809 \sin(\theta3) \sin(\theta2) q1p \\
& + 78.809 \cos(\theta3) \cos(\theta2) q1p) - 7.832223000 (\cos(\theta3) \sin(\theta2) q1p \\
& + \sin(\theta3) \cos(\theta2) q1p) (130 \sin(\theta3) \sin(\theta2) q1p \\
& - 130 \cos(\theta3) \cos(\theta2) q1p) - 7.832223000 \sin(\theta3) (-200 \cos(\theta2)^2 q1p^2 \\
& - 200 q2p^2 + \sin(\theta2) gz) + 7.832223000 \cos(\theta3) (200 q2pp \\
& + 200 \sin(\theta2) q1p^2 \cos(\theta2) + \cos(\theta2) gz) + (\cos(\theta3) \sin(\theta2) q1p \\
& + \sin(\theta3) \cos(\theta2) q1p) (-284.792 \sin(\theta3) \sin(\theta2) q1p \\
& + 284.792 \cos(\theta3) \cos(\theta2) q1p) - (-\sin(\theta3) \sin(\theta2) q1p \\
& + \cos(\theta3) \cos(\theta2) q1p) (282.163 \cos(\theta3) \sin(\theta2) q1p \\
& + 282.163 \sin(\theta3) \cos(\theta2) q1p)
\end{aligned}$$

WWW.BDIGITAL.ULA.VE

Anexo C

Programa en MATLAB ® con las ecuaciones obtenidas para el cálculo de la cinemática directa.

WWW.BDIGITAL.ULA.VE

Licencia Creative Commons:
Atribución - No Comercial - Compartir Igual 3.0 Venezuela
(CC BY-NC-SA 3.0 VE)

```

close all;
clear all;
clc;

% Cinemática directa del manipulador calculada manualmente

theta1 = input('Introduzca el valor de theta1 en radianes: '),
theta2 = input('Introduzca el valor de theta2 en radianes: '),
theta3 = input('Introduzca el valor de theta3 en radianes: '),

Nx = cos(theta1)*cos(theta2)*cos(theta3)-cos(theta1)*sin(theta2)*sin(theta3);
Ny = sin(theta1)*cos(theta2)*cos(theta3)-sin(theta1)*sin(theta2)*sin(theta3);
Nz = sin(theta2)*cos(theta3)+cos(theta2)*sin(theta3);

Ox = -cos(theta1)*cos(theta2)*sin(theta3)-cos(theta1)*sin(theta2)*cos(theta3);
Oy = -sin(theta1)*cos(theta2)*sin(theta3)-sin(theta1)*sin(theta2)*cos(theta3);
Oz = -sin(theta2)*sin(theta3)+cos(theta2)*cos(theta3);

Ax = sin(theta1);
Ay = -cos(theta1);
Az = 0;

Px = 130*cos(theta1)*cos(theta2)*cos(theta3)-130*cos(theta1)*sin(theta2)*...
    sin(theta3)+200*cos(theta1)*cos(theta2),
Py = 130*sin(theta1)*cos(theta2)*cos(theta3)-130*sin(theta1)*sin(theta2)*...
    sin(theta3)+200*sin(theta1)*cos(theta2),
Pz = 130*sin(theta2)*cos(theta3)+130*cos(theta2)*sin(theta3)+230+200*...
    sin(theta2),

Tdirecta = [[Nx, Ox, Ax, Px];[Ny, Oy, Ay, Py];[Nz, Oz, Az, Pz];[0,0,0,1]],

```

Anexo D

Programa en MATLAB ® que utiliza el “Robotic Toolbox for MATLAB” para el cálculo de las cinemática directa, cinemática inversa, Jacobianas directas.

WWW.BDIGITAL.ULA.VE

Licencia Creative Commons:
Atribución - No Comercial - Compartir Igual 3.0 Venezuela
(CC BY-NC-SA 3.0 VE)

```
close all;
```

```
clear all;
```

```
clc;
```

```
% Luis José Andueza Castro
```

```
% Especialización y Maestría en Ingeniería de Control y Automatización
```

```
% Escuela de Ingeniería de Sistemas
```

```
% Universidad de Los Andes
```

```
%
```

```
%
```

```
% Programa para el cálculo de los parámetros del manipulador robótico.
```

```
% Datos los parámetros de Denavit - Hartenberg
```

```
%
```

```
%
```

```
% Los Parámetro de Denavit - Hartenberg (según Barrientos et Al.)
```

```
% que se tienen son:
```

```
%
```

```
%
```

```
% |-----|
```

Articulación	Theta	d	a	Alfa
1	Theta 1	230	0	90
2	Theta 2	0	200	0
3	Theta 3	0	130	0

```
% |-----|
```

```
% | 1 | Theta 1 | 230 | 0 | 90 |
```

```
% |-----|
```

```
% | 2 | Theta 2 | 0 | 200 | 0 |
```

```
% |-----|
```

```
% | 3 | Theta 3 | 0 | 130 | 0 |
```

```
% |-----|
```

```
%
```

```
% Donde:
```

```
%
```

```
% Theta: Es el ángulo que forman los ejes  $x_{i-1}$  y  $x_i$  medido en un plano
```

```
% perpendicular al eje  $z_{i-1}$ , utilizando la regla de la mano derecha.
```

```
% Es un parámetro para articulaciones giratorias.
```

```
%
```

```
% d: Es la distancia a lo largo del eje  $z_{i-1}$  desde el origen del sistema de
```

```
% coordenadas  $(i-1)$ -ésimo hasta la intersección del eje  $z_{i-1}$  con el eje  $x_i$ .
```

```
% En articulaciones prismáticas es un parámetro variable.
```

Licencia Creative Commons:

Atribución - No Comercial - Compartir Igual 3.0 Venezuela
(CC BY-NC-SA 3.0 VE)

```

%
% a: Es la distancia a lo largo del eje xi que va desde la intersección del
% eje zi-1 con el eje xi hasta el origen del sistema i-ésimo, en el caso
% de articulaciones giratorias. En el caso de articulaciones prismáticas,
% se calcula como la distancia más corta entre los ejes z1-i y zi.
%
% Alfa: Es el ángulo de separación del eje zi-1 y el eje zi, medido en un
% plano perpendicular al eje xi, utilizando la regla de la mano derecha.
%
%
% El cálculo de la cinemática, directa e inversa. Las Jacobianas directas e
% inversas. La dinámica directa e inversa. Se harán con la ayuda de toolbox
% de MatLAB "Robotic Toolbox" creado por Peter I. Corke. CSRIO Australia.
%
% Para este TOOLBOX, se utiliza otra nomenclatura:
%
% Alfa: Es el ángulo de giro del eslabón.
% A: es la longitud del eslabón.
% Theta: Es el ángulo de rotación del eslabón.
% D: es el desplazamiento del eslabón.
% Sigma: Parámetro que indica si la articulación es prismática o rotacional
%
% Y se escriben en el orden: [Alfa, a, theta, d, sigma]
%
% Por lo que los parámetros Denavit - Hartenberg para el caso en estudio
% quedan como:
%
% -----
% | Articulación | Alfa | a | Theta | d | sigma |
% -----
% | 1 | pi/2 | 0 | 0 | 230 | 0 |
% -----
% | 2 | 0 | 200 | 0 | 0 | 0 |
% -----
% | 3 | 0 | 130 | 0 | 0 | 0 |
% -----

```

```

%
% Es de notar que el TOOLBOX utiliza radianes en vez de grados.
%
% Una vez obtenidos los parametros de Denavit - Hartenberg, se procede
% a definir los eslabones, donde cada eslabón estará definido con la letra
% L (link), donde L{1} es para el eslabón 1, L{2} para el eslabón 2 y así
% sucesivamente:

```

```
clear L;
```

```

% L{#} = número del eslabón y tipo
% L.m = masa del eslabón
% L.r = centro de masa del eslabón
% L.I = Tensores de inercia
% L.Jm = Inercia del motor que acciona el elemento
% L.G = Coeficiente de reducción del actuador del elemento
% L.B = Rozamiento viscoso del actuador del elemento
% L.Tc = Rozamiento de Coulomb en la dirección positiva y negativa del
% actuador del elemento

```

```
L{1} = link([pi/2, 0, 0, 230, 0]);
```

```
L{1}.m = 0.583;
```

```
L{1}.r = [0, -82, 0];
```

```
L{1}.I = [0, 0, 0];
```

```
L{1}.Jm = 0;
```

```
L{1}.G = 1;
```

```
L{1}.B = 0;
```

```
L{1}.Tc = [0, 0];
```

```
L{2} = link ([0, 200, 0, 0, 0]);
```

```
L{2}.m = 0.473;
```

```
L{2}.r = [-86.78, 0, 13];
```

```
L{2}.I = [0, 0, 0];
```

```
L{2}.Jm = 0;
```

```
L{2}.G = 1;
```

```
L{2}.B = 0;
```

```
L{2}.Tc = [0, 0];
```

```
L{3} = link ([0, 130, 0, 0, 0]);
```

```
L{3}.m = 0.153;
```

```
L{3}.r = [-78.8, 0, 0];
```

```
L{3}.I = [0, 0, 0];
```

```
L{3}.Jm = 0;
```

```
L{3}.G = 1;
```

```
L{3}.B = 0;
```

```
L{3}.Tc = [0, 0];
```

```
% Una vez definidos los eslabones del robot, se define el robot en sí.
```

```
Qz = [0 0 0]; % angulos todos en cero (0), posicion en L
```

```
BrazoRobotico = robot (L),
```

```
BrazoRobotico.name = 'Manipulador',
```

```
BrazoRobotico.gravity = [0, 0, 9810],
```

```
% BrazoRobotico.base = trans([0,0,0]),
```

```
% plot(BrazoRobotico, Qz),
```

```
% drivebot(BrazoRobotico),
```

```
% Se crearan diversas posiciones asignando valores de angulos para las
```

```
% articulaciones rotativas, para la cinematica directa.
```

```
% Qz, la primera posición será en posición inicial con todas articulaciones
```

```
% en cero grados.
```

```
Qz,
```

```
% Q1 la primera posicion será el brazo totalmente extendido, para lo cual.
```

```
% los valores articulares son los que se muestran a continuación
```

```
Q1 = [0, pi/2, 0], % recordando que los grados tienen que ser radianes
```

```
% Q2 será la posición donde el brazo se encuentra totalmente extendido a
```

% excepción del último eslabón que se encontrará formando un ángulo de 90
% grados con respecto al eslabón 2.

Q2 = [0, pi/2, -pi/2],

% Q3 será la misma posición anterior pero con el último eslabón formando
% los mismos 90 grados pero en el otro sentido.

Q3 = [0, pi/2, pi/2],

% Finalmente Q4 será la misma posición inicial pero en el otro sentido.

Q4 = [pi, 0, 0],

% Q5 posición "aleatoria"

Q5 = [0.7854, -0.5231, 1.0910],

% Q6 Posición "aleatoria"

Q6 = [0.7854, -1.4181, 1.9656],

% Q7 Posición "aleatoria"

Q7 = [-8.3176, -3.2222, 8.6734],

% Q8 Posición "aleatoria"

Q8 = [0.7854, 0.9334, -1.7058],

% Q9 Posición "aleatoria"

Q9 = [0.7854, 1.1445, -1.2378],

% Q10 Posición "aleatoria"

```
q10 = [pi/3, pi/3, pi/3],
```

```
%Una vez que se tienen las posiciones, se procede a probar la cinemática
%directa.
```

```
cdirectaQz = fkine(BrazoRobotico, Qz), % Cinemática directa para Qz
cdirectaQ1 = fkine(BrazoRobotico, Q1), % Cinemática directa para Q1
cdirectaQ2 = fkine(BrazoRobotico, Q2), % Cinemática directa para Q2
cdirectaQ3 = fkine(BrazoRobotico, Q3), % Cinemática directa para Q3
cdirectaQ4 = fkine(BrazoRobotico, Q4), % Cinemática directa para Q4
cdirectaQ5 = fkine(BrazoRobotico, Q5), % Cinemática directa para Q5
cdirectaQ6 = fkine(BrazoRobotico, Q6), % Cinemática directa para Q6
cdirectaQ7 = fkine(BrazoRobotico, Q7), % Cinemática directa para Q7
cdirectaQ8 = fkine(BrazoRobotico, Q8), % Cinemática directa para Q8
cdirectaQ9 = fkine(BrazoRobotico, Q9), % Cinemática directa para Q9
cdirectaQ10 = fkine(BrazoRobotico, Q10), % Cinemática directa para Q10
```

```
t = [0:1:10];
```

```
% cálculo de trayectorias rectas ctraj
```

```
% cqz = ctraj(Qz, Q1, t);
% cq1 = ctraj(Q1, Q2, t);
% cq2 = ctraj(Q2, Q3, t);
% cq3 = ctraj(Q3, Q4, t);
% cq4 = ctraj(Q4, Q5, t);
% cq5 = ctraj(Q5, Q6, t);
% cq6 = ctraj(Q6, Q7, t);
% cq7 = ctraj(Q7, Q8, t);
% cq8 = ctraj(Q8, Q9, t);
% cq9 = ctraj(Q9, Q10, t);
```

```
% cálculo de trayectorias jtraj
```

```
qz = jtraj(Qz, Q1, t);
q1 = jtraj(Q1, Q2, t);
q2 = jtraj(Q2, Q3, t);
```

```
q3 = jtraj(Q3, Q4, t);
q4 = jtraj(Q4, Q5, t);
q5 = jtraj(Q5, Q6, t);
q6 = jtraj(Q6, Q7, t);
q7 = jtraj(Q7, Q8, t);
q8 = jtraj(Q8, Q9, t);
q9 = jtraj(Q9, Q10, t);
```

```
plot(BrazoRobotico, qz),
plot(BrazoRobotico, q1),
plot(BrazoRobotico, q2),
plot(BrazoRobotico, q3),
plot(BrazoRobotico, q4),
plot(BrazoRobotico, q5),
plot(BrazoRobotico, q6),
plot(BrazoRobotico, q7),
plot(BrazoRobotico, q8),
plot(BrazoRobotico, q9),
```

```
%
JdirectaQz = jacob0(BrazoRobotico, Qz), % Jacobiana para Qz
JdirectaQ1 = jacob0(BrazoRobotico, Q1), % Jacobiana para Q1
JdirectaQ2 = jacob0(BrazoRobotico, Q2), % Jacobiana para Q2
JdirectaQ3 = jacob0(BrazoRobotico, Q3), % Jacobiana para Q3
JdirectaQ4 = jacob0(BrazoRobotico, Q4), % Jacobiana para Q4
JdirectaQ5 = jacob0(BrazoRobotico, Q5), % Jacobiana para Q5
JdirectaQ6 = jacob0(BrazoRobotico, Q6), % Jacobiana para Q6
JdirectaQ7 = jacob0(BrazoRobotico, Q7), % Jacobiana para Q7
JdirectaQ8 = jacob0(BrazoRobotico, Q8), % Jacobiana para Q8
JdirectaQ9 = jacob0(BrazoRobotico, Q9), % Jacobiana para Q9
JdirectaQ10 = jacob0(BrazoRobotico, Q10), % Jacobiana para Q10
```

```
% Cinematica inversa del robot
```

```
% utilizando las mismas matrices obtenidas de las cinematicas directas, se
% calcularán las cinemáticas inversas.
%
```

```
CinversaQz = ikine(BrazoRobotico, cdirectaQz), % Cinematica inversa para Qz
CinversaQ1 = ikine(BrazoRobotico, cdirectaQ1), % Cinematica inversa para Q1
CinversaQ2 = ikine(BrazoRobotico, cdirectaQ2), % Cinematica inversa para Q2
CinversaQ3 = ikine(BrazoRobotico, cdirectaQ3), % Cinematica inversa para Q3
% CinversaQ4 = ikine(BrazoRobotico, cdirectaQ4), % Cinematica inversa para
% Q4, esta solución no tiene solución, el programa no encuentra
% convergencia
CinversaQ5 = ikine(BrazoRobotico, cdirectaQ5), % Cinematica inversa para Q5
CinversaQ6 = ikine(BrazoRobotico, cdirectaQ6), % Cinematica inversa para Q6
CinversaQ7 = ikine(BrazoRobotico, cdirectaQ7), % Cinematica inversa para Q7
CinversaQ8 = ikine(BrazoRobotico, cdirectaQ8), % Cinematica inversa para Q8
CinversaQ9 = ikine(BrazoRobotico, cdirectaQ9), % Cinematica inversa para Q9
CinversaQ10 = ikine(BrazoRobotico, cdirectaQ10), % Cinematica inversa para Q10
```

WWW.BDIGITAL.ULA.VE

Anexo E

Programa en MATLAB ® con las ecuaciones obtenidas para el cálculo de la Jacobiana Directa.

WWW.BDIGITAL.ULA.VE

Licencia Creative Commons:
Atribución - No Comercial - Compartir Igual 3.0 Venezuela
(CC BY-NC-SA 3.0 VE)

```
close all;
clear all;
clc;

% Jacobiana directa

theta1 = input('Introduzca el valor de theta1 en radianes: '),
theta2 = input('Introduzca el valor de theta2 en radianes: '),
theta3 = input('Introduzca el valor de theta3 en radianes: '),

Jdirecta(1,1) = -
130*sin(theta1)*cos(theta2)*cos(theta3)+130*sin(theta1)*sin(theta2)*sin(theta3)-
200*sin(theta1)*cos(theta2);

Jdirecta(1,2) = -130*cos(theta1)*sin(theta2)*cos(theta3)-
130*cos(theta1)*cos(theta2)*sin(theta3)-200*sin(theta2)*cos(theta1);

Jdirecta(1,3) = -130*cos(theta1)*cos(theta2)*sin(theta3)-
130*cos(theta1)*sin(theta2)*cos(theta3);

Jdirecta(2,1) = 130*cos(theta1)*cos(theta2)*cos(theta3)-
130*cos(theta1)*sin(theta2)*sin(theta3)+200*cos(theta1)*cos(theta2);

Jdirecta(2,2) = -130*sin(theta1)*sin(theta2)*cos(theta3)-
130*sin(theta1)*cos(theta2)*sin(theta3)-200*sin(theta2)*sin(theta1);

Jdirecta(2,3) = -130*sin(theta1)*cos(theta2)*sin(theta3)-
130*sin(theta1)*sin(theta2)*cos(theta3);

Jdirecta(3,1) = 0;

Jdirecta(3,2) = 130*cos(theta2)*cos(theta3)-130*sin(theta2)*sin(theta3)+200*cos(theta2);

Jdirecta(3,3) = -130*sin(theta2)*sin(theta3)+130*cos(theta2)*cos(theta3);

Jdirecta,
```

Anexo F

Programa en MATLAB ® con las ecuaciones obtenidas para el cálculo de la Jacobiana Inversa, invirtiendo la Jacobiana Directa.

WWW.BDIGITAL.ULA.VE

Licencia Creative Commons:
Atribución - No Comercial - Compartir Igual 3.0 Venezuela
(CC BY-NC-SA 3.0 VE)

```
% Jacobiana inversa calculada a partir de la Jacobiana directa.
```

```
close all;
```

```
clear all;
```

```
clc;
```

```
% Cinemática inversa calculada a través de las matrices homogéneas
```

```
Px = input('Introduzca el valor de la coordenada en X: '),
```

```
Py = input('Introduzca el valor de la coordenada en Y: '),
```

```
Pz = input('Introduzca el valor de la coordenada en Z: '),
```

```
theta1 = atan(Py/Px);
```

```
a = (cos(theta1)*Px+sin(theta1)*Py);
```

```
b = (Pz-230);
```

```
c = (23100+a^2+b^2)/400; % (23100+a*a+b*b)/400;
```

```
theta2a = 2*atan((b+(sqrt(b^2-c^2+a^2)))/(a+c));
```

```
theta2b = 2*atan((b-(sqrt(b^2-c^2+a^2)))/(a+c));
```

```
theta3a = atan((b*cos(theta2a)-a*sin(theta2a))/(a*cos(theta2a)+b*sin(theta2a)-200));
```

```
theta3b = atan((b*cos(theta2b)-a*sin(theta2b))/(a*cos(theta2b)+b*sin(theta2b)-200));
```

```
[theta1 theta2a theta2b theta3a theta3b],
```

```
% Jacobiana inversa a partir de la Jacobiana directa
```

```
JinversaA(1,1) = 1/10*sin(theta1)/(13*sin(theta2a)*sin(theta3a)- ...  
20*cos(theta2a)-13*cos(theta2a)*cos(theta3a));
```

```
JinversaA(1,2) = -1/10*cos(theta1)/(13*sin(theta2a)*sin(theta3a)- ...  
20*cos(theta2a)-13*cos(theta2a)*cos(theta3a));
```

```
JinversaA(1,3) = 0;
```

```
JinversaA(2,1) = -1/200*(sin(theta2a)*sin(theta3a)-cos(theta2a)*cos(theta3a)) ...  
*cos(theta1)/sin(theta3a);
```

$$J_{\text{inversaA}}(2,2) = -1/200 * (\sin(\theta_{2a}) * \sin(\theta_{3a}) - \cos(\theta_{2a}) * \cos(\theta_{3a})) * \dots \\ \sin(\theta_{1}) / \sin(\theta_{3a});$$

$$J_{\text{inversaA}}(2,3) = 1/200 * (\sin(\theta_{2a}) * \cos(\theta_{3a}) + \cos(\theta_{2a}) * \sin(\theta_{3a})) / \dots \\ \sin(\theta_{3a});$$

$$J_{\text{inversaA}}(3,1) = 1/2600 * \cos(\theta_{1}) * (13 * \sin(\theta_{2a}) * \sin(\theta_{3a}) - 20 * \dots \\ \cos(\theta_{2a}) - 13 * \cos(\theta_{2a}) * \cos(\theta_{3a})) / \sin(\theta_{3a});$$

$$J_{\text{inversaA}}(3,2) = 1/2600 * \sin(\theta_{1}) * (13 * \sin(\theta_{2a}) * \sin(\theta_{3a}) - 20 * \dots \\ \cos(\theta_{2a}) - 13 * \cos(\theta_{2a}) * \cos(\theta_{3a})) / \sin(\theta_{3a});$$

$$J_{\text{inversaA}}(3,3) = -1/2600 * (20 * \sin(\theta_{2a}) + 13 * \sin(\theta_{2a}) * \cos(\theta_{3a}) + 13 * \dots \\ \cos(\theta_{2a}) * \sin(\theta_{3a})) / \sin(\theta_{3a});$$

J_{inversaA} ,

$$J_{\text{inversaB}}(1,1) = 1/10 * \sin(\theta_{1}) / (13 * \sin(\theta_{2b}) * \sin(\theta_{3b}) - \dots \\ 20 * \cos(\theta_{2b}) - 13 * \cos(\theta_{2b}) * \cos(\theta_{3b}));$$

$$J_{\text{inversaB}}(1,2) = -1/10 * \cos(\theta_{1}) / (13 * \sin(\theta_{2b}) * \sin(\theta_{3b}) - \dots \\ 20 * \cos(\theta_{2b}) - 13 * \cos(\theta_{2b}) * \cos(\theta_{3b}));$$

$$J_{\text{inversaB}}(1,3) = 0;$$

$$J_{\text{inversaB}}(2,1) = -1/200 * (\sin(\theta_{2b}) * \sin(\theta_{3b}) - \cos(\theta_{2b}) * \cos(\theta_{3b})) * \dots \\ * \cos(\theta_{1}) / \sin(\theta_{3b});$$

$$J_{\text{inversaB}}(2,2) = -1/200 * (\sin(\theta_{2b}) * \sin(\theta_{3b}) - \cos(\theta_{2b}) * \cos(\theta_{3b})) * \dots \\ \sin(\theta_{1}) / \sin(\theta_{3b});$$

$$J_{\text{inversaB}}(2,3) = 1/200 * (\sin(\theta_{2b}) * \cos(\theta_{3b}) + \cos(\theta_{2b}) * \sin(\theta_{3b})) / \dots \\ \sin(\theta_{3b});$$

$$J_{\text{inversaB}}(3,1) = 1/2600 * \cos(\theta_{1}) * (13 * \sin(\theta_{2b}) * \sin(\theta_{3b}) - 20 * \dots \\ \cos(\theta_{2b}) - 13 * \cos(\theta_{2b}) * \cos(\theta_{3b})) / \sin(\theta_{3b});$$

$$J_{\text{inversaB}}(3,2) = 1/2600 * \sin(\theta_{1}) * (13 * \sin(\theta_{2b}) * \sin(\theta_{3b}) - 20 * \dots \\ \cos(\theta_{2b}) - 13 * \cos(\theta_{2b}) * \cos(\theta_{3b})) / \sin(\theta_{3b});$$

$$J_{\text{inversaB}}(3,3) = -1/2600 * (20 * \sin(\theta_{2b}) + 13 * \sin(\theta_{2b}) * \cos(\theta_{3b}) + 13 * \dots \\ \cos(\theta_{2b}) * \sin(\theta_{3b})) / \sin(\theta_{3b});$$

J_{inversaB}

Anexo G

Programa en MATLAB ® con las ecuaciones obtenidas para el cálculo de la Jacobiana Inversa, a partir de las matrices homogéneas inversas.

WWW.BDIGITAL.ULA.VE

Licencia Creative Commons:
Atribución - No Comercial - Compartir Igual 3.0 Venezuela
(CC BY-NC-SA 3.0 VE)

```
% Jacobiana inversa calculada a partir de la cinemática inversa homogénea
```

```
close all;
```

```
clear all;
```

```
clc;
```

```
% Cinemática inversa calculada a través de las matrices homogéneas
```

```
Px = input('Introduzca el valor de la coordenada en X: '),
```

```
Py = input('Introduzca el valor de la coordenada en Y: '),
```

```
Pz = input('Introduzca el valor de la coordenada en Z: '),
```

```
theta1 = atan(Py/Px);
```

```
a = (cos(theta1)*Px+sin(theta1)*Py);
```

```
b = (Pz-230);
```

```
c = (23100+a^2+b^2)/400; % (23100+a*a+b*b)/400;
```

```
theta2a = 2*atan((b+(sqrt(b^2-c^2+a^2)))/(a+c));
```

```
theta2b = 2*atan((b-(sqrt(b^2-c^2+a^2)))/(a+c));
```

```
theta3a = atan((b*cos(theta2a)-a*sin(theta2a))/(a*cos(theta2a)+b*sin(theta2a)-200));
```

```
theta3b = atan((b*cos(theta2b)-a*sin(theta2b))/(a*cos(theta2b)+b*sin(theta2b)-200));
```

```
[theta1 theta2a theta2b theta3a theta3b],
```

```
% Jacobiana Inversa calculada a partir de la cinemática inversa homogénea.
```

```
% utilizando theta2a
```

```
JinversaA(1,1) = -Py/(Px^2*(1+Py^2/Px^2));
```

```
JinversaA(1,2) = 1/(Px*(1+Py^2/Px^2));
```

```
JinversaA(1,3) = 0;
```

```
JinversaA(2,1) = 0.2e1 * (((Pz - 230) ^ 2) - (0.231e3 / 0.4e1 + (((1 + Py ^ 2 / Px ^ 2) ^ (-0.1e1 / 0.2e1) * ...
```

Licencia Creative Commons:

Atribución - No Comercial - Compartir Igual 3.0 Venezuela
(CC BY-NC-SA 3.0 VE)

$$\begin{aligned}
& P_x + P_y^2 / P_x * (1 + P_y^2 / P_x^2) ^ (-0.1e1 / 0.2e1) ^ 2) / \dots \\
& 0.400e3 + ((P_z - 230) ^ 2) / 0.400e3) ^ 2 + (((1 + P_y^2 / P_x^2) ^ \dots \\
& (-0.1e1 / 0.2e1) * P_x + P_y^2 / P_x * (1 + P_y^2 / P_x^2) ^ (-0.1e1 / 0.2e1) ^ 2)) \\
& ^ \dots \\
& (-0.1e1 / 0.2e1) * (-0.231e3 / 0.4e1 + (((1 + P_y^2 / P_x^2) ^ \dots \\
& (-0.1e1 / 0.2e1) * P_x + P_y^2 / P_x * (1 + P_y^2 / P_x^2) ^ (-0.1e1 / 0.2e1) ^ 2) / \\
& \dots \\
& 0.400e3 + ((P_z - 230) ^ 2) / 0.400e3) * ((1 + P_y^2 / P_x^2) ^ (-0.1e1 / 0.2e1) * \\
& \dots \\
& P_x + P_y^2 / P_x * (1 + P_y^2 / P_x^2) ^ (-0.1e1 / 0.2e1) * ((1 + P_y^2 / P_x^2) \\
& ^ \dots \\
& (-0.3e1 / 0.2e1) / P_x^2 * P_y^2 + (1 + P_y^2 / P_x^2) ^ (-0.1e1 / 0.2e1) - P_y^2 \\
& \dots \\
& / P_x^2 * (1 + P_y^2 / P_x^2) ^ (-0.1e1 / 0.2e1) + P_y^4 / P_x^4 * \dots \\
& (1 + P_y^2 / P_x^2) ^ (-0.3e1 / 0.2e1)) / 0.100e3 + (2 * ((1 + P_y^2 / P_x^2) ^ \\
& \dots \\
& (-0.1e1 / 0.2e1) * P_x + P_y^2 / P_x * (1 + P_y^2 / P_x^2) ^ (-0.1e1 / 0.2e1) * \dots \\
& ((1 + P_y^2 / P_x^2) ^ (-0.3e1 / 0.2e1) / P_x^2 * P_y^2 + (1 + P_y^2 / P_x^2) ^ \\
& \dots \\
& (-0.1e1 / 0.2e1) - P_y^2 / P_x^2 * (1 + P_y^2 / P_x^2) ^ (-0.1e1 / 0.2e1) + \dots \\
& P_y^4 / P_x^4 * (1 + P_y^2 / P_x^2) ^ (-0.3e1 / 0.2e1)))) / (((1 + P_y^2 / P_x^2) \\
& ^ \dots \\
& (-0.1e1 / 0.2e1) * P_x) + (P_y^2 / P_x * (1 + P_y^2 / P_x^2) ^ (-0.1e1 / 0.2e1) + \\
& \dots \\
& 0.231e3 / 0.4e1 + (((1 + P_y^2 / P_x^2) ^ (-0.1e1 / 0.2e1) * P_x + P_y^2 / P_x * \dots \\
& (1 + P_y^2 / P_x^2) ^ (-0.1e1 / 0.2e1) ^ 2) / 0.400e3 + ((P_z - 230) ^ 2) / 0.400e3) \\
& / \dots \\
& 0.2e1 - (P_z - 0.230e3 + \text{sqrt}(((P_z - 230) ^ 2) - (0.231e3 / 0.4e1 + (((1 + P_y^2 / P_x \\
& ^ 2) ^ \dots \\
& (-0.1e1 / 0.2e1) * P_x + P_y^2 / P_x * (1 + P_y^2 / P_x^2) ^ (-0.1e1 / 0.2e1) ^ 2) / \\
& \dots \\
& * 0.400e3 + ((P_z - 230) ^ 2) / 0.400e3) ^ 2 + (((1 + P_y^2 / P_x^2) ^ (-0.1e1 / 0.2e1) \\
& * \dots \\
& P_x + P_y^2 / P_x * (1 + P_y^2 / P_x^2) ^ (-0.1e1 / 0.2e1) ^ 2))) / (((1 + P_y^2 / P_x \\
& ^ 2) ^ \dots \\
& (-0.1e1 / 0.2e1) * P_x) + (P_y^2 / P_x * (1 + P_y^2 / P_x^2) ^ (-0.1e1 / 0.2e1) + \\
& 0.231e3 / \dots \\
& 0.4e1 + (((1 + P_y^2 / P_x^2) ^ (-0.1e1 / 0.2e1) * P_x + P_y^2 / P_x * (1 + P_y^2 / P_x \\
& ^ 2) ^ \dots \\
& (-0.1e1 / 0.2e1) ^ 2) / 0.400e3 + ((P_z - 230) ^ 2) / 0.400e3) ^ 2 * \dots \\
& (((1 + P_y^2 / P_x^2) ^ (-0.3e1 / 0.2e1) / P_x^2 * P_y^2) + ((1 + P_y^2 / P_x^2) \\
& ^ \dots \\
& (-0.1e1 / 0.2e1) - (P_y^2 / P_x^2 * (1 + P_y^2 / P_x^2) ^ \dots \\
& (-0.1e1 / 0.2e1) + (P_y^4 / P_x^4 * (1 + P_y^2 / P_x^2) ^ (-0.3e1 / 0.2e1)) + \dots \\
& (((1 + P_y^2 / P_x^2) ^ (-0.1e1 / 0.2e1) * P_x + P_y^2 / P_x * \dots \\
& (1 + P_y^2 / P_x^2) ^ (-0.1e1 / 0.2e1) * ((1 + P_y^2 / P_x^2) ^ \dots
\end{aligned}$$

Licencia Creative Commons:

Atribución - No Comercial - Compartir Igual 3.0 Venezuela
(CC BY-NC-SA 3.0 VE)

$$\begin{aligned}
 & (-0.3e1 / 0.2e1) / Px \wedge 2 * Py \wedge 2 + (1 + Py \wedge 2 / Px \wedge 2) \wedge (-0.1e1 / 0.2e1) - \dots \\
 & Py \wedge 2 / Px \wedge 2 * (1 + Py \wedge 2 / Px \wedge 2) \wedge (-0.1e1 / 0.2e1) + Py \wedge 4 / Px \wedge 4 * \dots \\
 & (1 + Py \wedge 2 / Px \wedge 2) \wedge (-0.3e1 / 0.2e1))) / 0.200e3)) / (0.1e1 + \dots \\
 & (Pz - 0.230e3 + \text{sqrt}(((Pz - 230) \wedge 2) - (0.231e3 / 0.4e1 + (((1 + Py \wedge 2 / Px \wedge 2) \wedge \\
 & \dots \\
 & (-0.1e1 / 0.2e1) * Px + Py \wedge 2 / Px * (1 + Py \wedge 2 / Px \wedge 2) \wedge (-0.1e1 / 0.2e1)) \wedge 2) \\
 & \dots \\
 & / 0.400e3 + ((Pz - 230) \wedge 2) / 0.400e3) \wedge 2 + (((1 + Py \wedge 2 / Px \wedge 2) \wedge \dots \\
 & (-0.1e1 / 0.2e1) * Px + Py \wedge 2 / Px * (1 + Py \wedge 2 / Px \wedge 2) \wedge (-0.1e1 / 0.2e1)) \wedge 2))) \\
 & \wedge 2 \dots \\
 & / (((1 + Py \wedge 2 / Px \wedge 2) \wedge (-0.1e1 / 0.2e1) * Px) + (Py \wedge 2 / Px * \dots \\
 & (1 + Py \wedge 2 / Px \wedge 2) \wedge (-0.1e1 / 0.2e1)) + 0.231e3 / 0.4e1 + \dots \\
 & (((1 + Py \wedge 2 / Px \wedge 2) \wedge (-0.1e1 / 0.2e1) * Px + Py \wedge 2 / Px * \dots \\
 & (1 + Py \wedge 2 / Px \wedge 2) \wedge (-0.1e1 / 0.2e1)) \wedge 2) / 0.400e3 + ((Pz - 230) \wedge 2) \dots \\
 & / 0.400e3) \wedge 2);
 \end{aligned}$$

$$\begin{aligned}
 \text{JinversaA}(2,2) = & 0.2e1 * (((Pz - 230) \wedge 2) - (0.231e3 / 0.4e1 + \dots \\
 & (((1 + Py \wedge 2 / Px \wedge 2) \wedge (-0.1e1 / 0.2e1) * Px + Py \wedge 2 / Px * \dots \\
 & (1 + Py \wedge 2 / Px \wedge 2) \wedge (-0.1e1 / 0.2e1)) \wedge 2) / 0.400e3 + ((Pz - 230) \wedge 2) / \dots \\
 & 0.400e3) \wedge 2 + (((1 + Py \wedge 2 / Px \wedge 2) \wedge (-0.1e1 / 0.2e1) * Px + Py \wedge 2 / \dots \\
 & Px * (1 + Py \wedge 2 / Px \wedge 2) \wedge (-0.1e1 / 0.2e1)) \wedge 2) \wedge (-0.1e1 / 0.2e1) * \dots \\
 & (-0.231e3 / 0.4e1 + (((1 + Py \wedge 2 / Px \wedge 2) \wedge (-0.1e1 / 0.2e1) * Px + \dots \\
 & Py \wedge 2 / Px * (1 + Py \wedge 2 / Px \wedge 2) \wedge (-0.1e1 / 0.2e1)) \wedge 2) / 0.400e3 + \dots \\
 & ((Pz - 230) \wedge 2) / 0.400e3) * ((1 + Py \wedge 2 / Px \wedge 2) \wedge (-0.1e1 / 0.2e1) * \dots \\
 & Px + Py \wedge 2 / Px * (1 + Py \wedge 2 / Px \wedge 2) \wedge (-0.1e1 / 0.2e1)) * \dots \\
 & (-1 + Py \wedge 2 / Px \wedge 2) \wedge (-0.3e1 / 0.2e1) / Px * Py + 2 * Py / Px * \dots \\
 & \dots \\
 & (1 + Py \wedge 2 / Px \wedge 2) \wedge (-0.1e1 / 0.2e1) - Py \wedge 3 / Px \wedge 3 * (1 + Py \wedge 2 / Px \wedge 2) \wedge \\
 & \dots \\
 & (-0.3e1 / 0.2e1)) / 0.100e3 + (2 * ((1 + Py \wedge 2 / Px \wedge 2) \wedge (-0.1e1 / 0.2e1) * \dots \\
 & Px + Py \wedge 2 / Px * (1 + Py \wedge 2 / Px \wedge 2) \wedge (-0.1e1 / 0.2e1)) * \dots \\
 & (-1 + Py \wedge 2 / Px \wedge 2) \wedge (-0.3e1 / 0.2e1) / Px * Py + 2 * Py / Px * \dots \\
 & \dots \\
 & (1 + Py \wedge 2 / Px \wedge 2) \wedge (-0.1e1 / 0.2e1) - Py \wedge 3 / Px \wedge 3 * (1 + Py \wedge 2 / Px \wedge 2) \wedge \\
 & \dots \\
 & (-0.3e1 / 0.2e1)))) / (((1 + Py \wedge 2 / Px \wedge 2) \wedge (-0.1e1 / 0.2e1) * Px) + \dots \\
 & (Py \wedge 2 / Px * (1 + Py \wedge 2 / Px \wedge 2) \wedge (-0.1e1 / 0.2e1)) + 0.231e3 / 0.4e1 + \dots \\
 & (((1 + Py \wedge 2 / Px \wedge 2) \wedge (-0.1e1 / 0.2e1) * Px + Py \wedge 2 / Px * \dots \\
 & (1 + Py \wedge 2 / Px \wedge 2) \wedge (-0.1e1 / 0.2e1)) \wedge 2) / 0.400e3 + ((Pz - 230) \wedge 2) / \\
 & 0.400e3) \dots \\
 & / 0.2e1 - (Pz - 0.230e3 + \text{sqrt}(((Pz - 230) \wedge 2) - (0.231e3 / 0.4e1 + \dots
 \end{aligned}$$

$$\begin{aligned}
& (((1 + Py \wedge 2 / Px \wedge 2) \wedge (-0.1e1 / 0.2e1) * Px + Py \wedge 2 / Px * \dots \\
& (1 + Py \wedge 2 / Px \wedge 2) \wedge (-0.1e1 / 0.2e1)) \wedge 2) / 0.400e3 + ((Pz - 230) \wedge 2) / 0.400e3) \\
& \wedge 2 \dots \\
& + (((1 + Py \wedge 2 / Px \wedge 2) \wedge (-0.1e1 / 0.2e1) * Px + Py \wedge 2 / Px * \dots \\
& (1 + Py \wedge 2 / Px \wedge 2) \wedge (-0.1e1 / 0.2e1)) \wedge 2))) / (((1 + Py \wedge 2 / Px \wedge 2) \wedge \dots \\
& (-0.1e1 / 0.2e1) * Px) + (Py \wedge 2 / Px * (1 + Py \wedge 2 / Px \wedge 2) \wedge (-0.1e1 / 0.2e1)) + \\
& \dots \\
& 0.231e3 / 0.4e1 + (((1 + Py \wedge 2 / Px \wedge 2) \wedge (-0.1e1 / 0.2e1) * Px + \dots \\
& Py \wedge 2 / Px * (1 + Py \wedge 2 / Px \wedge 2) \wedge (-0.1e1 / 0.2e1)) \wedge 2) / 0.400e3 + \dots \\
& ((Pz - 230) \wedge 2) / 0.400e3) \wedge 2 * (-((1 + Py \wedge 2 / Px \wedge 2) \wedge \dots \\
& (-0.3e1 / 0.2e1) / Px * Py) + (2 * Py / Px * (1 + Py \wedge 2 / Px \wedge 2) \wedge \dots \\
& (-0.1e1 / 0.2e1)) - (Py \wedge 3 / Px \wedge 3 * (1 + Py \wedge 2 / Px \wedge 2) \wedge \dots \\
& (-0.3e1 / 0.2e1))) + (((1 + Py \wedge 2 / Px \wedge 2) \wedge (-0.1e1 / 0.2e1) * Px + \dots \\
& Py \wedge 2 / Px * (1 + Py \wedge 2 / Px \wedge 2) \wedge (-0.1e1 / 0.2e1)) * (-(1 + Py \wedge 2 / Px \wedge 2) \wedge \\
& \dots \\
& (-0.3e1 / 0.2e1) / Px * Py + 2 * Py / Px * (1 + Py \wedge 2 / Px \wedge 2) \wedge \dots \\
& (-0.1e1 / 0.2e1) - Py \wedge 3 / Px \wedge 3 * (1 + Py \wedge 2 / Px \wedge 2) \wedge \dots \\
& (-0.3e1 / 0.2e1))) / 0.200e3)) / (0.1e1 + (Pz - 0.230e3 + sqrt(((Pz - 230) \wedge 2) - \dots \\
& (0.231e3 / 0.4e1 + (((1 + Py \wedge 2 / Px \wedge 2) \wedge (-0.1e1 / 0.2e1) * Px + \dots \\
& Py \wedge 2 / Px * (1 + Py \wedge 2 / Px \wedge 2) \wedge (-0.1e1 / 0.2e1)) \wedge 2) / 0.400e3 + \dots \\
& ((Pz - 230) \wedge 2) / 0.400e3) \wedge 2 + (((1 + Py \wedge 2 / Px \wedge 2) \wedge (-0.1e1 / 0.2e1) * \dots \\
& Px + Py \wedge 2 / Px * (1 + Py \wedge 2 / Px \wedge 2) \wedge (-0.1e1 / 0.2e1)) \wedge 2))) \wedge 2 / \dots \\
& (((1 + Py \wedge 2 / Px \wedge 2) \wedge (-0.1e1 / 0.2e1) * Px) + (Py \wedge 2 / Px * \dots \\
& (1 + Py \wedge 2 / Px \wedge 2) \wedge (-0.1e1 / 0.2e1))) + 0.231e3 / 0.4e1 + \dots \\
& (((1 + Py \wedge 2 / Px \wedge 2) \wedge (-0.1e1 / 0.2e1) * Px + Py \wedge 2 / Px * \dots \\
& (1 + Py \wedge 2 / Px \wedge 2) \wedge (-0.1e1 / 0.2e1)) \wedge 2) / 0.400e3 + ((Pz - 230) \wedge 2) / \dots \\
& 0.400e3) \wedge 2);
\end{aligned}$$

$$\begin{aligned}
JinversaA(2,3) &= 0.2e1 * ((0.1e1 + (((Pz - 230) \wedge 2) - (0.231e3 / 0.4e1 \dots \\
& + (((1 + Py \wedge 2 / Px \wedge 2) \wedge (-0.1e1 / 0.2e1) * Px + Py \wedge 2 / Px * \dots \\
& (1 + Py \wedge 2 / Px \wedge 2) \wedge (-0.1e1 / 0.2e1)) \wedge 2) / 0.400e3 + ((Pz - 230) \wedge 2) / \dots \\
& 0.400e3) \wedge 2 + (((1 + Py \wedge 2 / Px \wedge 2) \wedge (-0.1e1 / 0.2e1) * Px + Py \wedge 2 \dots \\
& / Px * (1 + Py \wedge 2 / Px \wedge 2) \wedge (-0.1e1 / 0.2e1)) \wedge 2)) \wedge (-0.1e1 / 0.2e1) * \dots \\
& ((2 * Pz) - 0.460e3 - 0.2e1 * (0.231e3 / 0.4e1 + (((1 + Py \wedge 2 / Px \wedge 2) \wedge \dots \\
& (-0.1e1 / 0.2e1) * Px + Py \wedge 2 / Px * (1 + Py \wedge 2 / Px \wedge 2) \wedge (-0.1e1 / 0.2e1)) \wedge 2) / \\
& \dots \\
& 0.400e3 + ((Pz - 230) \wedge 2) / 0.400e3) * (Pz / 0.200e3 - 0.23e2 / 0.20e2)) / \dots \\
& 0.2e1) / (((1 + Py \wedge 2 / Px \wedge 2) \wedge (-0.1e1 / 0.2e1) * Px) + (Py \wedge 2 / Px * \dots
\end{aligned}$$

$$\begin{aligned}
& (1 + Py^2 / Px^2)^{-0.1e1 / 0.2e1} + 0.231e3 / 0.4e1 + \dots \\
& (((1 + Py^2 / Px^2)^{-0.1e1 / 0.2e1} * Px + Py^2 / Px * \dots \\
& (1 + Py^2 / Px^2)^{-0.1e1 / 0.2e1})^2 / 0.400e3 + \dots \\
& ((Pz - 230)^2 / 0.400e3) - (Pz - 0.230e3 + \sqrt{((Pz - 230)^2 - \dots \\
& (0.231e3 / 0.4e1 + (((1 + Py^2 / Px^2)^{-0.1e1 / 0.2e1} * \dots \\
& Px + Py^2 / Px * (1 + Py^2 / Px^2)^{-0.1e1 / 0.2e1}))^2 / \dots \\
& 0.400e3 + ((Pz - 230)^2 / 0.400e3)^2 + (((1 + Py^2 / Px^2)^{-0.1e1 / 0.2e1} * Px + Py^2 / Px * \dots \\
& (-0.1e1 / 0.2e1) * Px + Py^2 / Px * (1 + Py^2 / Px^2)^{-0.1e1 / 0.2e1})^2) / \dots \\
& (-0.1e1 / 0.2e1) * Px + Py^2 / Px * (1 + Py^2 / Px^2)^{-0.1e1 / 0.2e1} * Px) + \dots \\
& (Py^2 / Px * (1 + Py^2 / Px^2)^{-0.1e1 / 0.2e1}) + 0.231e3 / \dots \\
& 0.4e1 + (((1 + Py^2 / Px^2)^{-0.1e1 / 0.2e1} * Px + Py^2 / Px * \dots \\
& (1 + Py^2 / Px^2)^{-0.1e1 / 0.2e1})^2 / 0.400e3 + ((Pz - 230)^2 / \dots \\
& 0.400e3)^2 * (Pz / 0.200e3 - 0.23e2 / 0.20e2) / (0.1e1 + \dots \\
& (Pz - 0.230e3 + \sqrt{((Pz - 230)^2 - (0.231e3 / 0.4e1 + (((1 + Py^2 / Px^2)^{-0.1e1 / 0.2e1} * Px + Py^2 / Px * \dots \\
& \dots \\
& (-0.1e1 / 0.2e1) * Px + Py^2 / Px * (1 + Py^2 / Px^2)^{-0.1e1 / 0.2e1})^2 / 0.400e3 + ((Pz - 230)^2 / 0.400e3)^2 + \dots \\
& (((1 + Py^2 / Px^2)^{-0.1e1 / 0.2e1} * Px + Py^2 / Px * \dots \\
& (1 + Py^2 / Px^2)^{-0.1e1 / 0.2e1})^2) / (((1 + Py^2 / Px^2)^{-0.1e1 / 0.2e1} * Px) + (Py^2 / Px * (1 + Py^2 / Px^2)^{-0.1e1 / 0.2e1}) + \dots \\
& (-0.1e1 / 0.2e1) * Px) + (Py^2 / Px * (1 + Py^2 / Px^2)^{-0.1e1 / 0.2e1}) + 0.231e3 / 0.4e1 + (((1 + Py^2 / Px^2)^{-0.1e1 / 0.2e1} * Px + Py^2 / Px * \dots \\
& (-0.1e1 / 0.2e1) * Px + Py^2 / Px * (1 + Py^2 / Px^2)^{-0.1e1 / 0.2e1})^2 / 0.400e3 + ((Pz - 230)^2 / 0.400e3)^2);
\end{aligned}$$

$$\begin{aligned}
JinversaA(3,1) = & ((-0.2e1 * (Pz - 230) * \sin(0.2e1 * \operatorname{atan}((Pz - 0.230e3 + \dots \\
& + \sqrt{((Pz - 230)^2 - (0.231e3 / 0.4e1 + (((1 + Py^2 / Px^2)^{-0.1e1 / 0.2e1} * Px + Py^2 / Px * \dots \\
& (-0.1e1 / 0.2e1) * Px + Py^2 / Px * (1 + Py^2 / Px^2)^{-0.1e1 / 0.2e1})^2 / 0.400e3 + ((Pz - 230)^2 / 0.400e3)^2 + \dots \\
& (((1 + Py^2 / Px^2)^{-0.1e1 / 0.2e1} * Px + Py^2 / Px * \dots \\
& (1 + Py^2 / Px^2)^{-0.1e1 / 0.2e1})^2) / (((1 + Py^2 / Px^2)^{-0.1e1 / 0.2e1} * Px) + (Py^2 / Px * (1 + Py^2 / Px^2)^{-0.1e1 / 0.2e1}) + \dots \\
& (-0.1e1 / 0.2e1) * Px) + (Py^2 / Px * (1 + Py^2 / Px^2)^{-0.1e1 / 0.2e1}) + 0.231e3 / 0.4e1 + (((1 + Py^2 / Px^2)^{-0.1e1 / 0.2e1} * Px + Py^2 / Px * \dots \\
& (-0.1e1 / 0.2e1) * Px + Py^2 / Px * (1 + Py^2 / Px^2)^{-0.1e1 / 0.2e1})^2 / 0.400e3 + ((Pz - 230)^2 / 0.400e3)) * \dots \\
& (((Pz - 230)^2 - (0.231e3 / 0.4e1 + (((1 + Py^2 / Px^2)^{-0.1e1 / 0.2e1} * Px + Py^2 / Px * \dots \\
& (-0.1e1 / 0.2e1) * Px + Py^2 / Px * (1 + Py^2 / Px^2)^{-0.1e1 / 0.2e1})^2 / 0.400e3 + ((Pz - 230)^2 / 0.400e3)^2) / \dots \\
& (-0.1e1 / 0.2e1) * Px + Py^2 / Px * (1 + Py^2 / Px^2)^{-0.1e1 / 0.2e1})^2 / 0.400e3 + ((Pz - 230)^2 / 0.400e3)^2 + \dots
\end{aligned}$$

Licencia Creative Commons:

Atribución - No Comercial - Compartir Igual 3.0 Venezuela
(CC BY-NC-SA 3.0 VE)

$$\begin{aligned}
& (((1 + Py^2 / Px^2) \wedge (-0.1e1 / 0.2e1) * Px + Py^2 / Px * (1 + Py^2 / Px^2) \wedge \\
& \dots \\
& (-0.1e1 / 0.2e1)) \wedge 2)) \wedge (-0.1e1 / 0.2e1) * (-(0.231e3 / 0.4e1 + \dots \\
& (((1 + Py^2 / Px^2) \wedge (-0.1e1 / 0.2e1) * Px + Py^2 / Px * (1 + Py^2 / Px^2) \wedge \\
& \dots \\
& (-0.1e1 / 0.2e1)) \wedge 2) / 0.400e3 + ((Pz - 230) \wedge 2) / 0.400e3) * \dots \\
& ((1 + Py^2 / Px^2) \wedge (-0.1e1 / 0.2e1) * Px + Py^2 / Px * \dots \\
& (1 + Py^2 / Px^2) \wedge (-0.1e1 / 0.2e1)) * ((1 + Py^2 / Px^2) \wedge \dots \\
& (-0.3e1 / 0.2e1) / Px^2 * Py^2 + (1 + Py^2 / Px^2) \wedge \dots \\
& (-0.1e1 / 0.2e1) - Py^2 / Px^2 * (1 + Py^2 / Px^2) \wedge (-0.1e1 / 0.2e1) + \dots \\
& Py^4 / Px^4 * (1 + Py^2 / Px^2) \wedge (-0.3e1 / 0.2e1)) / 0.100e3 + \dots \\
& (2 * ((1 + Py^2 / Px^2) \wedge (-0.1e1 / 0.2e1) * Px + Py^2 / Px * \dots \\
& (1 + Py^2 / Px^2) \wedge (-0.1e1 / 0.2e1)) * ((1 + Py^2 / Px^2) \wedge \dots \\
& (-0.3e1 / 0.2e1) / Px^2 * Py^2 + (1 + Py^2 / Px^2) \wedge \dots \\
& (-0.1e1 / 0.2e1) - Py^2 / Px^2 * (1 + Py^2 / Px^2) \wedge \dots \\
& (-0.1e1 / 0.2e1) + Py^4 / Px^4 * (1 + Py^2 / Px^2) \wedge \dots \\
& (-0.3e1 / 0.2e1)))) / (((1 + Py^2 / Px^2) \wedge (-0.1e1 / 0.2e1) * Px) + \dots \\
& (Py^2 / Px * (1 + Py^2 / Px^2) \wedge (-0.1e1 / 0.2e1)) + 0.231e3 / 0.4e1 + \dots \\
& (((1 + Py^2 / Px^2) \wedge (-0.1e1 / 0.2e1) * Px + Py^2 / Px * \dots \\
& (1 + Py^2 / Px^2) \wedge (-0.1e1 / 0.2e1)) \wedge 2) / 0.400e3 + \dots \\
& ((Pz - 230) \wedge 2) / 0.400e3) / 0.2e1 - (Pz - 0.230e3 + \text{sqrt}(((Pz - 230) \wedge 2) - \dots \\
& (0.231e3 / 0.4e1 + (((1 + Py^2 / Px^2) \wedge (-0.1e1 / 0.2e1) * Px + Py^2 / Px * \dots \\
& (1 + Py^2 / Px^2) \wedge (-0.1e1 / 0.2e1)) \wedge 2) / 0.400e3 + \dots \\
& ((Pz - 230) \wedge 2) / 0.400e3) \wedge 2 + (((1 + Py^2 / Px^2) \wedge \dots \\
& (-0.1e1 / 0.2e1) * Px + Py^2 / Px * (1 + Py^2 / Px^2) \wedge \dots \\
& (-0.1e1 / 0.2e1)) \wedge 2))) / (((1 + Py^2 / Px^2) \wedge (-0.1e1 / 0.2e1) * Px) + \dots \\
& (Py^2 / Px * (1 + Py^2 / Px^2) \wedge (-0.1e1 / 0.2e1)) + 0.231e3 / 0.4e1 + \dots \\
& (((1 + Py^2 / Px^2) \wedge (-0.1e1 / 0.2e1) * Px + Py^2 / Px * \dots \\
& (1 + Py^2 / Px^2) \wedge (-0.1e1 / 0.2e1)) \wedge 2) / 0.400e3 + \dots \\
& ((Pz - 230) \wedge 2) / 0.400e3) \wedge 2 * (((1 + Py^2 / Px^2) \wedge \dots \\
& (-0.3e1 / 0.2e1) / Px^2 * Py^2) + ((1 + Py^2 / Px^2) \wedge \dots \\
& (-0.1e1 / 0.2e1)) - (Py^2 / Px^2 * (1 + Py^2 / Px^2) \wedge \dots \\
& (-0.1e1 / 0.2e1)) + (Py^4 / Px^4 * (1 + Py^2 / Px^2) \wedge \dots \\
& (-0.3e1 / 0.2e1)) + (((1 + Py^2 / Px^2) \wedge (-0.1e1 / 0.2e1) * \dots \\
& Px + Py^2 / Px * (1 + Py^2 / Px^2) \wedge (-0.1e1 / 0.2e1)) * \dots \\
& ((1 + Py^2 / Px^2) \wedge (-0.3e1 / 0.2e1) / Px^2 * Py^2 + \dots \\
& (1 + Py^2 / Px^2) \wedge (-0.1e1 / 0.2e1) - Py^2 / Px^2 * \dots
\end{aligned}$$

Licencia Creative Commons:

Atribución - No Comercial - Compartir Igual 3.0 Venezuela
(CC BY-NC-SA 3.0 VE)

$$\begin{aligned}
& (1 + Py^2 / Px^2)^{-0.1e1 / 0.2e1} + Py^4 / Px^4 * (1 + Py^2 / Px^2)^{-0.3e1 / 0.2e1} / 0.200e3) / (0.1e1 + (Pz - 0.230e3 + \dots \\
& \sqrt{((Pz - 230)^2 - (0.231e3 / 0.4e1 + ((1 + Py^2 / Px^2)^{-0.1e1 / 0.2e1} * Px + Py^2 / Px * (1 + Py^2 / Px^2)^{-0.1e1 / 0.2e1})^2) / 0.400e3 + ((Pz - 230)^2 / 0.400e3)^2 + \dots \\
& (((1 + Py^2 / Px^2)^{-0.1e1 / 0.2e1} * Px + Py^2 / Px * (1 + Py^2 / Px^2)^{-0.1e1 / 0.2e1})^2)) / \dots \\
& (((1 + Py^2 / Px^2)^{-0.1e1 / 0.2e1} * Px) + (Py^2 / Px * (1 + Py^2 / Px^2)^{-0.1e1 / 0.2e1})) + 0.231e3 / 0.4e1 + \dots \\
& (((1 + Py^2 / Px^2)^{-0.1e1 / 0.2e1} * Px + Py^2 / Px * (1 + Py^2 / Px^2)^{-0.1e1 / 0.2e1})^2) / 0.400e3 + \dots \\
& ((Pz - 230)^2 / 0.400e3)^2 - ((1 + Py^2 / Px^2)^{-0.1e1 / 0.2e1} * Px + Py^2 / Px * (1 + Py^2 / Px^2)^{-0.1e1 / 0.2e1})^2 + \dots \\
& (-0.1e1 / 0.2e1) - Py^2 / Px^2 * (1 + Py^2 / Px^2)^{-0.1e1 / 0.2e1} + Py^4 / Px^4 * (1 + Py^2 / Px^2)^{-0.1e1 / 0.2e1} + \dots \\
& (-0.3e1 / 0.2e1) * \sin(0.2e1 * \operatorname{atan}((Pz - 0.230e3 + \dots \\
& \sqrt{((Pz - 230)^2 - (0.231e3 / 0.4e1 + ((1 + Py^2 / Px^2)^{-0.1e1 / 0.2e1} * Px + Py^2 / Px * (1 + Py^2 / Px^2)^{-0.1e1 / 0.2e1})^2) / 0.400e3 + ((Pz - 230)^2 / 0.400e3)^2 + ((1 + Py^2 / Px^2)^{-0.1e1 / 0.2e1} * Px + Py^2 / Px * (1 + Py^2 / Px^2)^{-0.1e1 / 0.2e1})^2)) / (((1 + Py^2 / Px^2)^{-0.1e1 / 0.2e1} * Px) + \dots \\
& (Py^2 / Px * (1 + Py^2 / Px^2)^{-0.1e1 / 0.2e1}) + 0.231e3 / 0.4e1 + \dots \\
& (((1 + Py^2 / Px^2)^{-0.1e1 / 0.2e1} * Px + Py^2 / Px * (1 + Py^2 / Px^2)^{-0.1e1 / 0.2e1})^2) / 0.400e3 + \dots \\
& ((Pz - 230)^2 / 0.400e3)) - 0.2e1 * ((1 + Py^2 / Px^2)^{-0.1e1 / 0.2e1} * Px + Py^2 / Px * (1 + Py^2 / Px^2)^{-0.1e1 / 0.2e1}) * \cos(0.2e1 * \operatorname{atan}((Pz - 0.230e3 + \dots \\
& \sqrt{((Pz - 230)^2 - (0.231e3 / 0.4e1 + ((1 + Py^2 / Px^2)^{-0.1e1 / 0.2e1} * Px + Py^2 / Px * (1 + Py^2 / Px^2)^{-0.1e1 / 0.2e1})^2) / 0.400e3 + ((Pz - 230)^2 / 0.400e3)^2 + ((1 + Py^2 / Px^2)^{-0.1e1 / 0.2e1} * Px + Py^2 / Px * (1 + Py^2 / Px^2)^{-0.1e1 / 0.2e1})^2)) / (((1 + Py^2 / Px^2)^{-0.1e1 / 0.2e1} * Px) + \dots \\
& (-0.1e1 / 0.2e1) * Px) + (Py^2 / Px * (1 + Py^2 / Px^2)^{-0.1e1 / 0.2e1}) + 0.231e3 / 0.4e1 + ((1 + Py^2 / Px^2)^{-0.1e1 / 0.2e1} * Px + Py^2 / Px * (1 + Py^2 / Px^2)^{-0.1e1 / 0.2e1})^2) / 0.400e3 + ((Pz - 230)^2 / 0.400e3)) * \dots
\end{aligned}$$

Licencia Creative Commons:

Atribución - No Comercial - Compartir Igual 3.0 Venezuela
(CC BY-NC-SA 3.0 VE)

$$\begin{aligned}
 & (((Pz - 230) \wedge 2) - (0.231e3 / 0.4e1 + (((1 + Py \wedge 2 / Px \wedge 2) \wedge \dots \\
 & (-0.1e1 / 0.2e1) * Px + Py \wedge 2 / Px * (1 + Py \wedge 2 / Px \wedge 2) \wedge \dots \\
 & (-0.1e1 / 0.2e1)) \wedge 2) / 0.400e3 + ((Pz - 230) \wedge 2) / 0.400e3) \wedge 2 + \dots \\
 & (((1 + Py \wedge 2 / Px \wedge 2) \wedge (-0.1e1 / 0.2e1) * Px + Py \wedge 2 / Px * \dots \\
 & (1 + Py \wedge 2 / Px \wedge 2) \wedge (-0.1e1 / 0.2e1)) \wedge 2) \wedge (-0.1e1 / 0.2e1) * \dots \\
 & (-0.231e3 / 0.4e1 + (((1 + Py \wedge 2 / Px \wedge 2) \wedge (-0.1e1 / 0.2e1) * \dots \\
 & Px + Py \wedge 2 / Px * (1 + Py \wedge 2 / Px \wedge 2) \wedge (-0.1e1 / 0.2e1)) \wedge 2) / \dots \\
 & 0.400e3 + ((Pz - 230) \wedge 2) / 0.400e3) * ((1 + Py \wedge 2 / Px \wedge 2) \wedge \dots \\
 & (-0.1e1 / 0.2e1) * Px + Py \wedge 2 / Px * (1 + Py \wedge 2 / Px \wedge 2) \wedge \dots \\
 & (-0.1e1 / 0.2e1)) * ((1 + Py \wedge 2 / Px \wedge 2) \wedge (-0.3e1 / 0.2e1) / \dots \\
 & Px \wedge 2 * Py \wedge 2 + (1 + Py \wedge 2 / Px \wedge 2) \wedge (-0.1e1 / 0.2e1) - \dots \\
 & Py \wedge 2 / Px \wedge 2 * (1 + Py \wedge 2 / Px \wedge 2) \wedge (-0.1e1 / 0.2e1) + \dots \\
 & Py \wedge 4 / Px \wedge 4 * (1 + Py \wedge 2 / Px \wedge 2) \wedge (-0.3e1 / 0.2e1)) / 0.100e3 + \dots \\
 & (2 * ((1 + Py \wedge 2 / Px \wedge 2) \wedge (-0.1e1 / 0.2e1) * Px + Py \wedge 2 / Px * \dots \\
 & (1 + Py \wedge 2 / Px \wedge 2) \wedge (-0.1e1 / 0.2e1)) * ((1 + Py \wedge 2 / Px \wedge 2) \wedge \dots \\
 & (-0.3e1 / 0.2e1) / Px \wedge 2 * Py \wedge 2 + (1 + Py \wedge 2 / Px \wedge 2) \wedge \dots \\
 & (-0.1e1 / 0.2e1) - Py \wedge 2 / Px \wedge 2 * (1 + Py \wedge 2 / Px \wedge 2) \wedge \dots \\
 & (-0.1e1 / 0.2e1) + Py \wedge 4 / Px \wedge 4 * (1 + Py \wedge 2 / Px \wedge 2) \wedge (-0.3e1 / 0.2e1)))) / \dots \\
 & (((1 + Py \wedge 2 / Px \wedge 2) \wedge (-0.1e1 / 0.2e1) * Px) + (Py \wedge 2 / Px * \dots \\
 & (1 + Py \wedge 2 / Px \wedge 2) \wedge (-0.1e1 / 0.2e1)) + 0.231e3 / 0.4e1 + \dots \\
 & (((1 + Py \wedge 2 / Px \wedge 2) \wedge (-0.1e1 / 0.2e1) * Px + Py \wedge 2 / Px * \dots \\
 & (1 + Py \wedge 2 / Px \wedge 2) \wedge (-0.1e1 / 0.2e1)) \wedge 2) / 0.400e3 + \dots \\
 & ((Pz - 230) \wedge 2) / 0.400e3) / 0.2e1 - (Pz - 0.230e3 + sqrt(((Pz - 230) \wedge 2) - \dots \\
 & (0.231e3 / 0.4e1 + (((1 + Py \wedge 2 / Px \wedge 2) \wedge (-0.1e1 / 0.2e1) * Px + Py \wedge 2 \dots \\
 & / Px * (1 + Py \wedge 2 / Px \wedge 2) \wedge (-0.1e1 / 0.2e1)) \wedge 2) / 0.400e3 + \dots \\
 & ((Pz - 230) \wedge 2) / 0.400e3) \wedge 2 + (((1 + Py \wedge 2 / Px \wedge 2) \wedge \dots \\
 & (-0.1e1 / 0.2e1) * Px + Py \wedge 2 / Px * (1 + Py \wedge 2 / Px \wedge 2) \wedge \dots \\
 & (-0.1e1 / 0.2e1)) \wedge 2))) / (((1 + Py \wedge 2 / Px \wedge 2) \wedge (-0.1e1 / 0.2e1) * Px) + \dots \\
 & (Py \wedge 2 / Px * (1 + Py \wedge 2 / Px \wedge 2) \wedge (-0.1e1 / 0.2e1)) + 0.231e3 / 0.4e1 + \dots \\
 & (((1 + Py \wedge 2 / Px \wedge 2) \wedge (-0.1e1 / 0.2e1) * Px + Py \wedge 2 / Px * \dots \\
 & (1 + Py \wedge 2 / Px \wedge 2) \wedge (-0.1e1 / 0.2e1)) \wedge 2) / 0.400e3 + \dots \\
 & ((Pz - 230) \wedge 2) / 0.400e3) \wedge 2 * (((1 + Py \wedge 2 / Px \wedge 2) \wedge \dots \\
 & (-0.3e1 / 0.2e1) / Px \wedge 2 * Py \wedge 2) + ((1 + Py \wedge 2 / Px \wedge 2) \wedge \dots \\
 & (-0.1e1 / 0.2e1)) - (Py \wedge 2 / Px \wedge 2 * (1 + Py \wedge 2 / Px \wedge 2) \wedge \dots \\
 & (-0.1e1 / 0.2e1)) + (Py \wedge 4 / Px \wedge 4 * (1 + Py \wedge 2 / Px \wedge 2) \wedge \dots \\
 & (-0.3e1 / 0.2e1)) + (((1 + Py \wedge 2 / Px \wedge 2) \wedge (-0.1e1 / 0.2e1) * \dots
 \end{aligned}$$

$$\begin{aligned}
& P_x + P_y^2 / P_x * (1 + P_y^2 / P_x^2)^{(-0.1e1 / 0.2e1)} * \dots \\
& ((1 + P_y^2 / P_x^2)^{(-0.3e1 / 0.2e1)} / P_x^2 * P_y^2 + \dots \\
& (1 + P_y^2 / P_x^2)^{(-0.1e1 / 0.2e1)} - P_y^2 / P_x^2 * \dots \\
& (1 + P_y^2 / P_x^2)^{(-0.1e1 / 0.2e1)} + P_y^4 / P_x^4 * \dots \\
& (1 + P_y^2 / P_x^2)^{(-0.3e1 / 0.2e1)}) / 0.200e3)) / \dots \\
& (0.1e1 + (P_z - 0.230e3 + \text{sqrt}(((P_z - 230)^2) - (0.231e3 / 0.4e1 + \dots \\
& (((1 + P_y^2 / P_x^2)^{(-0.1e1 / 0.2e1)} * P_x + P_y^2 / P_x * \dots \\
& (1 + P_y^2 / P_x^2)^{(-0.1e1 / 0.2e1)})^2) / 0.400e3 + \dots \\
& ((P_z - 230)^2) / 0.400e3)^2 + (((1 + P_y^2 / P_x^2)^{(-0.1e1 / 0.2e1)} * P_x + P_y^2 / P_x * (1 + P_y^2 / P_x^2)^{(-0.1e1 / 0.2e1)})^2)))^2 / (((1 + P_y^2 / P_x^2)^{(-0.1e1 / 0.2e1)} * P_x) + (P_y^2 / P_x * (1 + P_y^2 / P_x^2)^{(-0.1e1 / 0.2e1)})) + 0.231e3 / 0.4e1 + (((1 + P_y^2 / P_x^2)^{(-0.1e1 / 0.2e1)} * P_x + P_y^2 / P_x * (1 + P_y^2 / P_x^2)^{(-0.1e1 / 0.2e1)})^2) / 0.400e3 + ((P_z - 230)^2) / 0.400e3)^2)) / \dots \\
& (((1 + P_y^2 / P_x^2)^{(-0.1e1 / 0.2e1)} * P_x + P_y^2 / P_x * \dots \\
& (1 + P_y^2 / P_x^2)^{(-0.1e1 / 0.2e1)}) * \cos(0.2e1 * \text{atan}((P_z - 0.230e3 + \dots \\
& \text{sqrt}(((P_z - 230)^2) - (0.231e3 / 0.4e1 + (((1 + P_y^2 / P_x^2)^{(-0.1e1 / 0.2e1)} * P_x + P_y^2 / P_x * (1 + P_y^2 / P_x^2)^{(-0.1e1 / 0.2e1)})^2) / 0.400e3 + ((P_z - 230)^2) / 0.400e3)^2 + \dots \\
& (((1 + P_y^2 / P_x^2)^{(-0.1e1 / 0.2e1)} * P_x + P_y^2 / P_x * \dots \\
& (1 + P_y^2 / P_x^2)^{(-0.1e1 / 0.2e1)})^2))) / (((1 + P_y^2 / P_x^2)^{(-0.1e1 / 0.2e1)} * P_x) + (P_y^2 / P_x * (1 + P_y^2 / P_x^2)^{(-0.1e1 / 0.2e1)})) + 0.231e3 / 0.4e1 + (((1 + P_y^2 / P_x^2)^{(-0.1e1 / 0.2e1)} * P_x + P_y^2 / P_x * (1 + P_y^2 / P_x^2)^{(-0.1e1 / 0.2e1)})^2) / 0.400e3 + ((P_z - 230)^2) / 0.400e3))) + \dots \\
& (P_z - 230) * \sin(0.2e1 * \text{atan}((P_z - 0.230e3 + \text{sqrt}(((P_z - 230)^2) - \dots \\
& (0.231e3 / 0.4e1 + (((1 + P_y^2 / P_x^2)^{(-0.1e1 / 0.2e1)} * P_x + \dots \\
& P_y^2 / P_x * (1 + P_y^2 / P_x^2)^{(-0.1e1 / 0.2e1)})^2) / 0.400e3 + \dots \\
& ((P_z - 230)^2) / 0.400e3)^2 + (((1 + P_y^2 / P_x^2)^{(-0.1e1 / 0.2e1)} * P_x + P_y^2 / P_x * (1 + P_y^2 / P_x^2)^{(-0.1e1 / 0.2e1)})^2))) / \dots \\
& (((1 + P_y^2 / P_x^2)^{(-0.1e1 / 0.2e1)} * P_x) + (P_y^2 / P_x * \dots \\
& (1 + P_y^2 / P_x^2)^{(-0.1e1 / 0.2e1)}) + 0.231e3 / 0.4e1 + \dots \\
& (((1 + P_y^2 / P_x^2)^{(-0.1e1 / 0.2e1)} * P_x + P_y^2 / P_x * \dots \\
& (1 + P_y^2 / P_x^2)^{(-0.1e1 / 0.2e1)})^2) / 0.400e3 + \dots \\
& ((P_z - 230)^2) / 0.400e3))) - 0.200e3) - ((P_z - 230) * \cos(0.2e1 * \dots
\end{aligned}$$

Licencia Creative Commons:

Atribución - No Comercial - Compartir Igual 3.0 Venezuela
(CC BY-NC-SA 3.0 VE)

$$\begin{aligned}
& \text{atan}((Pz - 0.230e3 + \text{sqrt}(((Pz - 230) \wedge 2) - (0.231e3 / 0.4e1 + \dots \\
& (((1 + Py \wedge 2 / Px \wedge 2) \wedge (-0.1e1 / 0.2e1) * Px + Py \wedge 2 / Px * \dots \\
& (1 + Py \wedge 2 / Px \wedge 2) \wedge (-0.1e1 / 0.2e1)) \wedge 2) / 0.400e3 + \dots \\
& ((Pz - 230) \wedge 2) / 0.400e3) \wedge 2 + (((1 + Py \wedge 2 / Px \wedge 2) \wedge \dots \\
& (-0.1e1 / 0.2e1) * Px + Py \wedge 2 / Px * (1 + Py \wedge 2 / Px \wedge 2) \wedge \dots \\
& (-0.1e1 / 0.2e1)) \wedge 2))) / (((1 + Py \wedge 2 / Px \wedge 2) \wedge (-0.1e1 / 0.2e1) * Px) + \dots \\
& (Py \wedge 2 / Px * (1 + Py \wedge 2 / Px \wedge 2) \wedge (-0.1e1 / 0.2e1)) + 0.231e3 / 0.4e1 + \dots \\
& (((1 + Py \wedge 2 / Px \wedge 2) \wedge (-0.1e1 / 0.2e1) * Px + Py \wedge 2 / Px * \dots \\
& (1 + Py \wedge 2 / Px \wedge 2) \wedge (-0.1e1 / 0.2e1)) \wedge 2) / 0.400e3 + \dots \\
& ((Pz - 230) \wedge 2) / 0.400e3))) - ((1 + Py \wedge 2 / Px \wedge 2) \wedge \dots \\
& (-0.1e1 / 0.2e1) * Px + Py \wedge 2 / Px * (1 + Py \wedge 2 / Px \wedge 2) \wedge \dots \\
& (-0.1e1 / 0.2e1)) * \sin(0.2e1 * \text{atan}((Pz - 0.230e3 + \dots \\
& \text{sqrt}(((Pz - 230) \wedge 2) - (0.231e3 / 0.4e1 + (((1 + Py \wedge 2 / Px \wedge 2) \wedge \dots \\
& (-0.1e1 / 0.2e1) * Px + Py \wedge 2 / Px * (1 + Py \wedge 2 / Px \wedge 2) \wedge \dots \\
& (-0.1e1 / 0.2e1)) \wedge 2) / 0.400e3 + ((Pz - 230) \wedge 2) / 0.400e3) \wedge 2 + \dots \\
& (((1 + Py \wedge 2 / Px \wedge 2) \wedge (-0.1e1 / 0.2e1) * Px + Py \wedge 2 / Px * \dots \\
& (1 + Py \wedge 2 / Px \wedge 2) \wedge (-0.1e1 / 0.2e1)) \wedge 2))) / (((1 + Py \wedge 2 / Px \wedge 2) \wedge \dots \\
& (-0.1e1 / 0.2e1) * Px) + (Py \wedge 2 / Px * (1 + Py \wedge 2 / Px \wedge 2) \wedge \dots \\
& (-0.1e1 / 0.2e1)) + 0.231e3 / 0.4e1 + (((1 + Py \wedge 2 / Px \wedge 2) \wedge \dots \\
& (-0.1e1 / 0.2e1) * Px + Py \wedge 2 / Px * (1 + Py \wedge 2 / Px \wedge 2) \wedge \dots \\
& (-0.1e1 / 0.2e1)) \wedge 2) / 0.400e3 + ((Pz - 230) \wedge 2) / 0.400e3))) / \dots \\
& (((1 + Py \wedge 2 / Px \wedge 2) \wedge (-0.1e1 / 0.2e1) * Px + Py \wedge 2 / Px * \dots \\
& (1 + Py \wedge 2 / Px \wedge 2) \wedge (-0.1e1 / 0.2e1)) * \cos(0.2e1 * \dots \\
& \text{atan}((Pz - 0.230e3 + \text{sqrt}(((Pz - 230) \wedge 2) - (0.231e3 / 0.4e1 + \dots \\
& (((1 + Py \wedge 2 / Px \wedge 2) \wedge (-0.1e1 / 0.2e1) * Px + Py \wedge 2 / Px * \dots \\
& (1 + Py \wedge 2 / Px \wedge 2) \wedge (-0.1e1 / 0.2e1)) \wedge 2) / 0.400e3 + \dots \\
& ((Pz - 230) \wedge 2) / 0.400e3) \wedge 2 + (((1 + Py \wedge 2 / Px \wedge 2) \wedge \dots \\
& (-0.1e1 / 0.2e1) * Px + Py \wedge 2 / Px * (1 + Py \wedge 2 / Px \wedge 2) \wedge \dots \\
& (-0.1e1 / 0.2e1)) \wedge 2))) / (((1 + Py \wedge 2 / Px \wedge 2) \wedge (-0.1e1 / 0.2e1) * Px) + \dots \\
& (Py \wedge 2 / Px * (1 + Py \wedge 2 / Px \wedge 2) \wedge (-0.1e1 / 0.2e1)) + 0.231e3 / 0.4e1 + \dots \\
& (((1 + Py \wedge 2 / Px \wedge 2) \wedge (-0.1e1 / 0.2e1) * Px + Py \wedge 2 / Px * \dots \\
& (1 + Py \wedge 2 / Px \wedge 2) \wedge (-0.1e1 / 0.2e1)) \wedge 2) / 0.400e3 + \dots \\
& ((Pz - 230) \wedge 2) / 0.400e3))) + (Pz - 230) * \sin(0.2e1 * \dots \\
& \text{atan}((Pz - 0.230e3 + \text{sqrt}(((Pz - 230) \wedge 2) - (0.231e3 / 0.4e1 + \dots \\
& (((1 + Py \wedge 2 / Px \wedge 2) \wedge (-0.1e1 / 0.2e1) * Px + Py \wedge 2 / Px * \dots \\
& (1 + Py \wedge 2 / Px \wedge 2) \wedge (-0.1e1 / 0.2e1)) \wedge 2) / 0.400e3 + \dots
\end{aligned}$$

Licencia Creative Commons:

Atribución - No Comercial - Compartir Igual 3.0 Venezuela
(CC BY-NC-SA 3.0 VE)

$$\begin{aligned}
& ((Pz - 230) \wedge 2) / 0.400e3 \wedge 2 + (((1 + Py \wedge 2 / Px \wedge 2) \wedge \dots \\
& (-0.1e1 / 0.2e1) * Px + Py \wedge 2 / Px * (1 + Py \wedge 2 / Px \wedge 2) \wedge \dots \\
& (-0.1e1 / 0.2e1) \wedge 2))) / (((1 + Py \wedge 2 / Px \wedge 2) \wedge (-0.1e1 / 0.2e1) * Px) + \dots \\
& (Py \wedge 2 / Px * (1 + Py \wedge 2 / Px \wedge 2) \wedge (-0.1e1 / 0.2e1)) + 0.231e3 / 0.4e1 + \dots \\
& (((1 + Py \wedge 2 / Px \wedge 2) \wedge (-0.1e1 / 0.2e1) * Px + Py \wedge 2 / Px * \dots \\
& (1 + Py \wedge 2 / Px \wedge 2) \wedge (-0.1e1 / 0.2e1)) \wedge 2) / 0.400e3 + \dots \\
& ((Pz - 230) \wedge 2) / 0.400e3))) - 0.200e3 \wedge 2 * (((1 + Py \wedge 2 / Px \wedge 2) \wedge \dots \\
& (-0.3e1 / 0.2e1) / Px \wedge 2 * Py \wedge 2 + (1 + Py \wedge 2 / Px \wedge 2) \wedge \dots \\
& (-0.1e1 / 0.2e1) - Py \wedge 2 / Px \wedge 2 * (1 + Py \wedge 2 / Px \wedge 2) \wedge \dots \\
& (-0.1e1 / 0.2e1) + Py \wedge 4 / Px \wedge 4 * (1 + Py \wedge 2 / Px \wedge 2) \wedge \dots \\
& (-0.3e1 / 0.2e1)) * \cos(0.2e1 * \operatorname{atan}((Pz - 0.230e3 + \operatorname{sqrt}(((Pz - 230) \wedge 2) - \dots \\
& (0.231e3 / 0.4e1 + (((1 + Py \wedge 2 / Px \wedge 2) \wedge (-0.1e1 / 0.2e1) * \dots \\
& Px + Py \wedge 2 / Px * (1 + Py \wedge 2 / Px \wedge 2) \wedge (-0.1e1 / 0.2e1)) \wedge 2) / \dots \\
& 0.400e3 + ((Pz - 230) \wedge 2) / 0.400e3) \wedge 2 + (((1 + Py \wedge 2 / Px \wedge 2) \wedge \dots \\
& (-0.1e1 / 0.2e1) * Px + Py \wedge 2 / Px * (1 + Py \wedge 2 / Px \wedge 2) \wedge \dots \\
& (-0.1e1 / 0.2e1) \wedge 2))) / (((1 + Py \wedge 2 / Px \wedge 2) \wedge (-0.1e1 / 0.2e1) * Px) + \dots \\
& (Py \wedge 2 / Px * (1 + Py \wedge 2 / Px \wedge 2) \wedge (-0.1e1 / 0.2e1)) + 0.231e3 / 0.4e1 + \dots \\
& (((1 + Py \wedge 2 / Px \wedge 2) \wedge (-0.1e1 / 0.2e1) * Px + Py \wedge 2 / Px * \dots \\
& (1 + Py \wedge 2 / Px \wedge 2) \wedge (-0.1e1 / 0.2e1)) \wedge 2) / 0.400e3 + \dots \\
& ((Pz - 230) \wedge 2) / 0.400e3))) - 0.2e1 * ((1 + Py \wedge 2 / Px \wedge 2) \wedge \dots \\
& (-0.1e1 / 0.2e1) * Px + Py \wedge 2 / Px * (1 + Py \wedge 2 / Px \wedge 2) \wedge \dots \\
& (-0.1e1 / 0.2e1)) * \sin(0.2e1 * \operatorname{atan}((Pz - 0.230e3 + \operatorname{sqrt}(((Pz - 230) \wedge 2) - \dots \\
& (0.231e3 / 0.4e1 + (((1 + Py \wedge 2 / Px \wedge 2) \wedge (-0.1e1 / 0.2e1) * \dots \\
& Px + Py \wedge 2 / Px * (1 + Py \wedge 2 / Px \wedge 2) \wedge (-0.1e1 / 0.2e1)) \wedge 2) / \dots \\
& 0.400e3 + ((Pz - 230) \wedge 2) / 0.400e3) \wedge 2 + (((1 + Py \wedge 2 / Px \wedge 2) \wedge \dots \\
& (-0.1e1 / 0.2e1) * Px + Py \wedge 2 / Px * (1 + Py \wedge 2 / Px \wedge 2) \wedge \dots \\
& (-0.1e1 / 0.2e1) \wedge 2))) / (((1 + Py \wedge 2 / Px \wedge 2) \wedge (-0.1e1 / 0.2e1) * Px) + \dots \\
& (Py \wedge 2 / Px * (1 + Py \wedge 2 / Px \wedge 2) \wedge (-0.1e1 / 0.2e1)) + 0.231e3 / 0.4e1 + \dots \\
& (((1 + Py \wedge 2 / Px \wedge 2) \wedge (-0.1e1 / 0.2e1) * Px + Py \wedge 2 / Px * \dots \\
& (1 + Py \wedge 2 / Px \wedge 2) \wedge (-0.1e1 / 0.2e1)) \wedge 2) / 0.400e3 + \dots \\
& ((Pz - 230) \wedge 2) / 0.400e3))) * (((Pz - 230) \wedge 2) - \dots \\
& (0.231e3 / 0.4e1 + (((1 + Py \wedge 2 / Px \wedge 2) \wedge (-0.1e1 / 0.2e1) * \dots \\
& Px + Py \wedge 2 / Px * (1 + Py \wedge 2 / Px \wedge 2) \wedge (-0.1e1 / 0.2e1)) \wedge 2) / \dots \\
& 0.400e3 + ((Pz - 230) \wedge 2) / 0.400e3) \wedge 2 + (((1 + Py \wedge 2 / Px \wedge 2) \wedge \dots \\
& (-0.1e1 / 0.2e1) * Px + Py \wedge 2 / Px * (1 + Py \wedge 2 / Px \wedge 2) \wedge \dots \\
& (-0.1e1 / 0.2e1) \wedge 2)) \wedge (-0.1e1 / 0.2e1) * (-0.231e3 / 0.4e1 + \dots
\end{aligned}$$

Licencia Creative Commons:

Atribución - No Comercial - Compartir Igual 3.0 Venezuela
(CC BY-NC-SA 3.0 VE)

$$\begin{aligned}
&(((1 + Py^2 / Px^2) \wedge (-0.1e1 / 0.2e1) * Px + Py^2 / Px * \dots \\
&(1 + Py^2 / Px^2) \wedge (-0.1e1 / 0.2e1)) \wedge 2) / 0.400e3 + \dots \\
&((Pz - 230) \wedge 2) / 0.400e3) * ((1 + Py^2 / Px^2) \wedge \dots \\
&(-0.1e1 / 0.2e1) * Px + Py^2 / Px * (1 + Py^2 / Px^2) \wedge \dots \\
&(-0.1e1 / 0.2e1)) * ((1 + Py^2 / Px^2) \wedge (-0.3e1 / 0.2e1) / \dots \\
&Px^2 * Py^2 + (1 + Py^2 / Px^2) \wedge (-0.1e1 / 0.2e1) - \dots \\
&Py^2 / Px^2 * (1 + Py^2 / Px^2) \wedge (-0.1e1 / 0.2e1) + \dots \\
&Py^4 / Px^4 * (1 + Py^2 / Px^2) \wedge (-0.3e1 / 0.2e1)) / \dots \\
&0.100e3 + (2 * ((1 + Py^2 / Px^2) \wedge (-0.1e1 / 0.2e1) * Px + Py^2 / \dots \\
&Px * (1 + Py^2 / Px^2) \wedge (-0.1e1 / 0.2e1)) * ((1 + Py^2 / Px^2) \wedge \dots \\
&(-0.3e1 / 0.2e1) / Px^2 * Py^2 + (1 + Py^2 / Px^2) \wedge \dots \\
&(-0.1e1 / 0.2e1) - Py^2 / Px^2 * (1 + Py^2 / Px^2) \wedge \dots \\
&(-0.1e1 / 0.2e1) + Py^4 / Px^4 * (1 + Py^2 / Px^2) \wedge \dots \\
&(-0.3e1 / 0.2e1)))) / (((1 + Py^2 / Px^2) \wedge (-0.1e1 / 0.2e1) * \dots \\
&Px) + (Py^2 / Px * (1 + Py^2 / Px^2) \wedge (-0.1e1 / 0.2e1)) + \dots \\
&0.231e3 / 0.4e1 + (((1 + Py^2 / Px^2) \wedge (-0.1e1 / 0.2e1) * \dots \\
&Px + Py^2 / Px * (1 + Py^2 / Px^2) \wedge (-0.1e1 / 0.2e1)) \wedge 2) / \dots \\
&0.400e3 + ((Pz - 230) \wedge 2) / 0.400e3) / 0.2e1 - (Pz - 0.230e3 + \dots \\
&\text{sqrt}(((Pz - 230) \wedge 2) - (0.231e3 / 0.4e1 + (((1 + Py^2 / Px^2) \wedge \dots \\
&(-0.1e1 / 0.2e1) * Px + Py^2 / Px * (1 + Py^2 / Px^2) \wedge \dots \\
&(-0.1e1 / 0.2e1)) \wedge 2) / 0.400e3 + ((Pz - 230) \wedge 2) / 0.400e3) \wedge 2 + \dots \\
&(((1 + Py^2 / Px^2) \wedge (-0.1e1 / 0.2e1) * Px + Py^2 / Px * \dots \\
&(1 + Py^2 / Px^2) \wedge (-0.1e1 / 0.2e1)) \wedge 2))) / (((1 + Py^2 / Px^2) \wedge \dots \\
&(-0.1e1 / 0.2e1) * Px) + (Py^2 / Px * (1 + Py^2 / Px^2) \wedge \dots \\
&(-0.1e1 / 0.2e1)) + 0.231e3 / 0.4e1 + (((1 + Py^2 / Px^2) \wedge \dots \\
&(-0.1e1 / 0.2e1) * Px + Py^2 / Px * (1 + Py^2 / Px^2) \wedge \dots \\
&(-0.1e1 / 0.2e1)) \wedge 2) / 0.400e3 + ((Pz - 230) \wedge 2) / 0.400e3) \wedge 2 * \dots \\
&(((1 + Py^2 / Px^2) \wedge (-0.3e1 / 0.2e1) / Px^2 * Py^2) + \dots \\
&((1 + Py^2 / Px^2) \wedge (-0.1e1 / 0.2e1)) - (Py^2 / Px^2 * \dots \\
&(1 + Py^2 / Px^2) \wedge (-0.1e1 / 0.2e1)) + (Py^4 / Px^4 * \dots \\
&(1 + Py^2 / Px^2) \wedge (-0.3e1 / 0.2e1)) + (((1 + Py^2 / Px^2) \wedge \dots \\
&(-0.1e1 / 0.2e1) * Px + Py^2 / Px * (1 + Py^2 / Px^2) \wedge \dots \\
&(-0.1e1 / 0.2e1)) * ((1 + Py^2 / Px^2) \wedge (-0.3e1 / 0.2e1) / \dots \\
&Px^2 * Py^2 + (1 + Py^2 / Px^2) \wedge (-0.1e1 / 0.2e1) - Py^2 / \dots \\
&Px^2 * (1 + Py^2 / Px^2) \wedge (-0.1e1 / 0.2e1) + Py^4 / Px^4 * \dots \\
&(1 + Py^2 / Px^2) \wedge (-0.3e1 / 0.2e1))) / 0.200e3)) / (0.1e1 + \dots
\end{aligned}$$

$$\begin{aligned}
 & (Pz - 0.230e3 + \text{sqrt}(((Pz - 230) \wedge 2) - (0.231e3 / 0.4e1 + (((1 + \dots \\
 & Py \wedge 2 / Px \wedge 2) \wedge (-0.1e1 / 0.2e1) * Px + Py \wedge 2 / Px * (1 + Py \wedge 2 / \dots \\
 & Px \wedge 2) \wedge (-0.1e1 / 0.2e1)) \wedge 2) / 0.400e3 + ((Pz - 230) \wedge 2) / 0.400e3) \wedge \dots \\
 & 2 + (((1 + Py \wedge 2 / Px \wedge 2) \wedge (-0.1e1 / 0.2e1) * Px + Py \wedge 2 / Px * \dots \\
 & (1 + Py \wedge 2 / Px \wedge 2) \wedge (-0.1e1 / 0.2e1)) \wedge 2))) \wedge 2 / (((1 + Py \wedge 2 / \dots \\
 & Px \wedge 2) \wedge (-0.1e1 / 0.2e1) * Px) + (Py \wedge 2 / Px * (1 + Py \wedge 2 / Px \wedge 2) \wedge \dots \\
 & (-0.1e1 / 0.2e1))) + 0.231e3 / 0.4e1 + (((1 + Py \wedge 2 / Px \wedge 2) \wedge \dots \\
 & (-0.1e1 / 0.2e1) * Px + Py \wedge 2 / Px * (1 + Py \wedge 2 / Px \wedge 2) \wedge \dots \\
 & (-0.1e1 / 0.2e1)) \wedge 2) / 0.400e3 + ((Pz - 230) \wedge 2) / 0.400e3) \wedge 2) + \dots \\
 & 0.2e1 * (Pz - 230) * \text{cos}(0.2e1 * \text{atan}((Pz - 0.230e3 + \dots \\
 & \text{sqrt}(((Pz - 230) \wedge 2) - (0.231e3 / 0.4e1 + (((1 + Py \wedge 2 / Px \wedge 2) \wedge \dots \\
 & (-0.1e1 / 0.2e1) * Px + Py \wedge 2 / Px * (1 + Py \wedge 2 / Px \wedge 2) \wedge \dots \\
 & (-0.1e1 / 0.2e1)) \wedge 2) / 0.400e3 + ((Pz - 230) \wedge 2) / 0.400e3) \wedge 2 + \dots \\
 & (((1 + Py \wedge 2 / Px \wedge 2) \wedge (-0.1e1 / 0.2e1) * Px + Py \wedge 2 / Px * \dots \\
 & (1 + Py \wedge 2 / Px \wedge 2) \wedge (-0.1e1 / 0.2e1)) \wedge 2))) / (((1 + Py \wedge 2 / Px \wedge 2) \wedge \dots \\
 & (-0.1e1 / 0.2e1) * Px) + (Py \wedge 2 / Px * (1 + Py \wedge 2 / Px \wedge 2) \wedge \dots \\
 & (-0.1e1 / 0.2e1))) + 0.231e3 / 0.4e1 + (((1 + Py \wedge 2 / Px \wedge 2) \wedge \dots \\
 & (-0.1e1 / 0.2e1) * Px + Py \wedge 2 / Px * (1 + Py \wedge 2 / Px \wedge 2) \wedge \dots \\
 & (-0.1e1 / 0.2e1)) \wedge 2) / 0.400e3 + ((Pz - 230) \wedge 2) / 0.400e3))) * \dots \\
 & (((Pz - 230) \wedge 2) - (0.231e3 / 0.4e1 + (((1 + Py \wedge 2 / Px \wedge 2) \wedge \dots \\
 & (-0.1e1 / 0.2e1) * Px + Py \wedge 2 / Px * (1 + Py \wedge 2 / Px \wedge 2) \wedge \dots \\
 & (-0.1e1 / 0.2e1)) \wedge 2) / 0.400e3 + ((Pz - 230) \wedge 2) / 0.400e3) \wedge 2 + \dots \\
 & (((1 + Py \wedge 2 / Px \wedge 2) \wedge (-0.1e1 / 0.2e1) * Px + Py \wedge 2 / Px * \dots \\
 & (1 + Py \wedge 2 / Px \wedge 2) \wedge (-0.1e1 / 0.2e1)) \wedge 2)) \wedge (-0.1e1 / 0.2e1) * \dots \\
 & (-0.231e3 / 0.4e1 + (((1 + Py \wedge 2 / Px \wedge 2) \wedge (-0.1e1 / 0.2e1) * Px + \dots \\
 & Py \wedge 2 / Px * (1 + Py \wedge 2 / Px \wedge 2) \wedge (-0.1e1 / 0.2e1)) \wedge 2) / 0.400e3 + \dots \\
 & ((Pz - 230) \wedge 2) / 0.400e3) * ((1 + Py \wedge 2 / Px \wedge 2) \wedge (-0.1e1 / 0.2e1) * \dots \\
 & Px + Py \wedge 2 / Px * (1 + Py \wedge 2 / Px \wedge 2) \wedge (-0.1e1 / 0.2e1)) * ((1 + Py \wedge \dots \\
 & 2 / Px \wedge 2) \wedge (-0.3e1 / 0.2e1) / Px \wedge 2 * Py \wedge 2 + (1 + Py \wedge 2 / Px \wedge 2) \wedge \dots \\
 & (-0.1e1 / 0.2e1) - Py \wedge 2 / Px \wedge 2 * (1 + Py \wedge 2 / Px \wedge 2) \wedge \dots \\
 & (-0.1e1 / 0.2e1) + Py \wedge 4 / Px \wedge 4 * (1 + Py \wedge 2 / Px \wedge 2) \wedge \dots \\
 & (-0.3e1 / 0.2e1)) / 0.100e3 + (2 * ((1 + Py \wedge 2 / Px \wedge 2) \wedge \dots \\
 & (-0.1e1 / 0.2e1) * Px + Py \wedge 2 / Px * (1 + Py \wedge 2 / Px \wedge 2) \wedge \dots \\
 & (-0.1e1 / 0.2e1)) * ((1 + Py \wedge 2 / Px \wedge 2) \wedge (-0.3e1 / 0.2e1) / \dots \\
 & Px \wedge 2 * Py \wedge 2 + (1 + Py \wedge 2 / Px \wedge 2) \wedge (-0.1e1 / 0.2e1) - Py \wedge 2 / \dots \\
 & Px \wedge 2 * (1 + Py \wedge 2 / Px \wedge 2) \wedge (-0.1e1 / 0.2e1) + Py \wedge 4 / Px \wedge 4 * \dots
 \end{aligned}$$

$$\frac{(1 + Py^2 / Px^2)^{-0.3e1 / 0.2e1}}{((-0.1e1 / 0.2e1) * Px + (Py^2 / Px * (1 + Py^2 / Px^2)^{-0.1e1 / 0.2e1}) + 0.231e3 / 0.4e1 + ((1 + Py^2 / Px^2)^{-0.1e1 / 0.2e1} * Px + Py^2 / Px * (1 + Py^2 / Px^2)^{-0.1e1 / 0.2e1})^2) / 0.4}$$

WWW.BDIGITAL.ULA.VE

Anexo H

Programa en MATLAB® con las ecuaciones obtenidas para el cálculo de la Jacobiana Inversa, a partir de la cinemática inversa calculada geoméricamente.

WWW.BDIGITAL.ULA.VE

Licencia Creative Commons:
Atribución - No Comercial - Compartir Igual 3.0 Venezuela
(CC BY-NC-SA 3.0 VE)

```

% Cálculo de la jacobiana nversa a partir de la cinemática inversa
% geométrica

close all;
clear all;
clc;

% Cinemática inversa calculada geoméricamente

Px = input('Introduzca el valor de la coordenada en x: ');
Py = input('Introduzca el valor de la coordenada en y: ');
Pz = input('Introduzca el valor de la coordenada en z: ');

D = (Px^2+Py^2+(Pz-230)^2-200^2-130^2)/(2*200*130);
theta1 = atan(Py/Px),
theta3Up = atan(sqrt(1-D^2)/D),
theta3Down = atan(-sqrt(1-D^2)/D),
theta2a = atan((Pz-230)/sqrt(Px^2+Py^2))-
atan((130*sin(theta3Up))/(200+130*cos(theta3Up))),
theta2b = atan((Pz-230)/sqrt(Px^2+Py^2))-
atan((130*sin(theta3Down))/(200+130*cos(theta3Down))),

% Cálculo de la jacobiana inversa utilizando theta3up

JinvGeomA(1,1) = -Py/(Px^2*(1+Py^2/Px^2));

JinvGeomA(1,2) = 1/(Px*(1+Py^2/Px^2));

JinvGeomA(1,3) = 0;

JinvGeomA(2,1) = -((Pz - 230) * (Px ^ 2 + Py ^ 2) ^ (-0.3e1 / 0.2e1) * ...
Px / (1 + (Pz - 230) ^ 2 / (Px ^ 2 + Py ^ 2))) - (0.65e2 * ((2688000000 - ...
2 * Px ^ 2 * Py ^ 2 - 2 * Px ^ 2 * Pz ^ 2 + 920 * Px ^ 2 * Pz - 2 * ...
Py ^ 2 * Pz ^ 2 + 8000 * Py ^ 2 - 3680000 * Pz + 8000 * Px ^ 2 - 203600 * ...
Pz ^ 2 - Px ^ 4 - Py ^ 4 - Pz ^ 4 + 920 * Py ^ 2 * Pz + 920 * Pz ^ 3) ^ ...
(-0.1e1 / 0.2e1)) / ((Px ^ 2) / 0.52000e5 + (Py ^ 2) / 0.52000e5 + ...
((Pz - 230) ^ 2) / 0.52000e5 - 0.569e3 / 0.520e3) * (0.2704000000e10 + ...

```

$$\begin{aligned}
 & (2688000000 - 2 * Px \wedge 2 * Py \wedge 2 - 2 * Px \wedge 2 * Pz \wedge 2 + 920 * Px \wedge 2 * \dots \\
 & Pz - 2 * Py \wedge 2 * Pz \wedge 2 + 8000 * Py \wedge 2 - 3680000 * Pz + 8000 * Px \wedge 2 - \dots \\
 & 203600 * Pz \wedge 2 - Px \wedge 4 - Py \wedge 4 - Pz \wedge 4 + 920 * Py \wedge 2 * Pz + 920 * \dots \\
 & Pz \wedge 3) / ((Px \wedge 2) / 0.52000e5 + (Py \wedge 2) / 0.52000e5 + ((Pz - 230) \wedge 2) / \dots \\
 & 0.52000e5 - 0.569e3 / 0.520e3) \wedge 2) \wedge (-0.1e1 / 0.2e1) / (0.200e3 + \dots \\
 & 0.6760000e7 * (0.2704000000e10 + (2688000000 - 2 * Px \wedge 2 * Py \wedge 2 - 2 * \dots \\
 & Px \wedge 2 * Pz \wedge 2 + 920 * Px \wedge 2 * Pz - 2 * Py \wedge 2 * Pz \wedge 2 + 8000 * Py \wedge 2 - \dots \\
 & 3680000 * Pz + 8000 * Px \wedge 2 - 203600 * Pz \wedge 2 - Px \wedge 4 - Py \wedge 4 - Pz \wedge 4 + \dots \\
 & 920 * Py \wedge 2 * Pz + 920 * Pz \wedge 3) / ((Px \wedge 2) / 0.52000e5 + (Py \wedge 2) / \dots \\
 & 0.52000e5 + ((Pz - 230) \wedge 2) / 0.52000e5 - 0.569e3 / 0.520e3) \wedge 2) \wedge \dots \\
 & (-0.1e1 / 0.2e1)) * (-4 * Px * Py \wedge 2 - 4 * Px * Pz \wedge 2 + 1840 * Px * \dots \\
 & Pz + 16000 * Px - 4 * Px \wedge 3) - \text{sqrt}((2688000000 - 2 * Px \wedge 2 * Py \wedge 2 - \dots \\
 & 2 * Px \wedge 2 * Pz \wedge 2 + 920 * Px \wedge 2 * Pz - 2 * Py \wedge 2 * Pz \wedge 2 + 8000 * \dots \\
 & Py \wedge 2 - 3680000 * Pz + 8000 * Px \wedge 2 - 203600 * Pz \wedge 2 - Px \wedge 4 - Py \wedge 4 - \dots \\
 & Pz \wedge 4 + 920 * Py \wedge 2 * Pz + 920 * Pz \wedge 3)) / ((Px \wedge 2) / 0.52000e5 + \dots \\
 & (Py \wedge 2) / 0.52000e5 + ((Pz - 230) \wedge 2) / 0.52000e5 - 0.569e3 / \dots \\
 & 0.520e3) \wedge 2 * (0.2704000000e10 + (2688000000 - 2 * Px \wedge 2 * Py \wedge 2 - \dots \\
 & 2 * Px \wedge 2 * Pz \wedge 2 + 920 * Px \wedge 2 * Pz - 2 * Py \wedge 2 * Pz \wedge 2 + 8000 * \dots \\
 & Py \wedge 2 - 3680000 * Pz + 8000 * Px \wedge 2 - 203600 * Pz \wedge 2 - Px \wedge 4 - Py \wedge 4 - \dots \\
 & Pz \wedge 4 + 920 * Py \wedge 2 * Pz + 920 * Pz \wedge 3) / ((Px \wedge 2) / 0.52000e5 + \dots \\
 & (Py \wedge 2) / 0.52000e5 + ((Pz - 230) \wedge 2) / 0.52000e5 - 0.569e3 / \dots \\
 & 0.520e3) \wedge 2) \wedge (-0.1e1 / 0.2e1) / (0.200e3 + 0.6760000e7 * \dots \\
 & (0.2704000000e10 + (2688000000 - 2 * Px \wedge 2 * Py \wedge 2 - 2 * Px \wedge 2 * \dots \\
 & Pz \wedge 2 + 920 * Px \wedge 2 * Pz - 2 * Py \wedge 2 * Pz \wedge 2 + 8000 * Py \wedge 2 - \dots \\
 & 3680000 * Pz + 8000 * Px \wedge 2 - 203600 * Pz \wedge 2 - Px \wedge 4 - Py \wedge 4 - \dots \\
 & Pz \wedge 4 + 920 * Py \wedge 2 * Pz + 920 * Pz \wedge 3) / ((Px \wedge 2) / 0.52000e5 + \dots \\
 & (Py \wedge 2) / 0.52000e5 + ((Pz - 230) \wedge 2) / 0.52000e5 - 0.569e3 / \dots \\
 & 0.520e3) \wedge 2) \wedge (-0.1e1 / 0.2e1)) * Px / 0.200e3 - 0.65e2 * \dots \\
 & \text{sqrt}((2688000000 - 2 * Px \wedge 2 * Py \wedge 2 - 2 * Px \wedge 2 * Pz \wedge 2 + 920 * \dots \\
 & Px \wedge 2 * Pz - 2 * Py \wedge 2 * Pz \wedge 2 + 8000 * Py \wedge 2 - 3680000 * Pz + \dots \\
 & 8000 * Px \wedge 2 - 203600 * Pz \wedge 2 - Px \wedge 4 - Py \wedge 4 - Pz \wedge 4 + 920 * \dots \\
 & Py \wedge 2 * Pz + 920 * Pz \wedge 3)) / ((Px \wedge 2) / 0.52000e5 + (Py \wedge 2) / \dots \\
 & 0.52000e5 + ((Pz - 230) \wedge 2) / 0.52000e5 - 0.569e3 / 0.520e3) * \dots \\
 & (0.2704000000e10 + (2688000000 - 2 * Px \wedge 2 * Py \wedge 2 - 2 * Px \wedge 2 * \dots \\
 & Pz \wedge 2 + 920 * Px \wedge 2 * Pz - 2 * Py \wedge 2 * Pz \wedge 2 + 8000 * Py \wedge 2 - \dots \\
 & 3680000 * Pz + 8000 * Px \wedge 2 - 203600 * Pz \wedge 2 - Px \wedge 4 - Py \wedge 4 - \dots
 \end{aligned}$$

WWW.BDIGITAL.ULA.VE

$$\begin{aligned}
 & Pz \wedge 4 + 920 * Py \wedge 2 * Pz + 920 * Pz \wedge 3) / ((Px \wedge 2) / 0.52000e5 + \dots \\
 & (Py \wedge 2) / 0.52000e5 + ((Pz - 230) \wedge 2) / 0.52000e5 - 0.569e3 / \dots \\
 & 0.520e3) \wedge 2) \wedge (-0.3e1 / 0.2e1) / (0.200e3 + 0.6760000e7 * \dots \\
 & (0.2704000000e10 + (2688000000 - 2 * Px \wedge 2 * Py \wedge 2 - 2 * Px \wedge 2 * \dots \\
 & Pz \wedge 2 + 920 * Px \wedge 2 * Pz - 2 * Py \wedge 2 * Pz \wedge 2 + 8000 * Py \wedge 2 - \dots \\
 & 3680000 * Pz + 8000 * Px \wedge 2 - 203600 * Pz \wedge 2 - Px \wedge 4 - Py \wedge 4 - \dots \\
 & Pz \wedge 4 + 920 * Py \wedge 2 * Pz + 920 * Pz \wedge 3) / ((Px \wedge 2) / 0.52000e5 + \dots \\
 & (Py \wedge 2) / 0.52000e5 + ((Pz - 230) \wedge 2) / 0.52000e5 - 0.569e3 / \dots \\
 & 0.520e3) \wedge 2) \wedge (-0.1e1 / 0.2e1)) * ((-4 * Px * Py \wedge 2 - 4 * Px * \dots \\
 & Pz \wedge 2 + 1840 * Px * Pz + 16000 * Px - 4 * Px \wedge 3) / ((Px \wedge 2) / \dots \\
 & 0.52000e5 + (Py \wedge 2) / 0.52000e5 + ((Pz - 230) \wedge 2) / 0.52000e5 - \dots \\
 & 0.569e3 / 0.520e3) \wedge 2 - (2688000000 - 2 * Px \wedge 2 * Py \wedge 2 - 2 * \dots \\
 & Px \wedge 2 * Pz \wedge 2 + 920 * Px \wedge 2 * Pz - 2 * Py \wedge 2 * Pz \wedge 2 + 8000 * \dots \\
 & Py \wedge 2 - 3680000 * Pz + 8000 * Px \wedge 2 - 203600 * Pz \wedge 2 - Px \wedge 4 - \dots \\
 & Py \wedge 4 - Pz \wedge 4 + 920 * Py \wedge 2 * Pz + 920 * Pz \wedge 3) / ((Px \wedge 2) / \dots \\
 & 0.52000e5 + (Py \wedge 2) / 0.52000e5 + ((Pz - 230) \wedge 2) / 0.52000e5 - \dots \\
 & 0.569e3 / 0.520e3) \wedge 3 * Px / 0.13000e5) + 0.439400000e9 * \dots \\
 & \text{sqrt}((2688000000 - 2 * Px \wedge 2 * Py \wedge 2 - 2 * Px \wedge 2 * Pz \wedge 2 + 920 * \dots \\
 & Px \wedge 2 * Pz - 2 * Py \wedge 2 * Pz \wedge 2 + 8000 * Py \wedge 2 - 3680000 * Pz + \dots \\
 & 8000 * Px \wedge 2 - 203600 * Pz \wedge 2 - Px \wedge 4 - Py \wedge 4 - Pz \wedge 4 + 920 * \dots \\
 & Py \wedge 2 * Pz + 920 * Pz \wedge 3)) / ((Px \wedge 2) / 0.52000e5 + (Py \wedge 2) / \dots \\
 & 0.52000e5 + ((Pz - 230) \wedge 2) / 0.52000e5 - 0.569e3 / 0.520e3) / \dots \\
 & (0.2704000000e10 + (2688000000 - 2 * Px \wedge 2 * Py \wedge 2 - 2 * Px \wedge 2 * \dots \\
 & Pz \wedge 2 + 920 * Px \wedge 2 * Pz - 2 * Py \wedge 2 * Pz \wedge 2 + 8000 * Py \wedge 2 - \dots \\
 & 3680000 * Pz + 8000 * Px \wedge 2 - 203600 * Pz \wedge 2 - Px \wedge 4 - Py \wedge 4 - \dots \\
 & Pz \wedge 4 + 920 * Py \wedge 2 * Pz + 920 * Pz \wedge 3) / ((Px \wedge 2) / 0.52000e5 + \dots \\
 & (Py \wedge 2) / 0.52000e5 + ((Pz - 230) \wedge 2) / 0.52000e5 - 0.569e3 / \dots \\
 & 0.520e3) \wedge 2) \wedge 2 / (0.200e3 + 0.6760000e7 * (0.2704000000e10 + \dots \\
 & (2688000000 - 2 * Px \wedge 2 * Py \wedge 2 - 2 * Px \wedge 2 * Pz \wedge 2 + 920 * \dots \\
 & Px \wedge 2 * Pz - 2 * Py \wedge 2 * Pz \wedge 2 + 8000 * Py \wedge 2 - 3680000 * Pz + \dots \\
 & 8000 * Px \wedge 2 - 203600 * Pz \wedge 2 - Px \wedge 4 - Py \wedge 4 - Pz \wedge 4 + 920 * \dots \\
 & Py \wedge 2 * Pz + 920 * Pz \wedge 3) / ((Px \wedge 2) / 0.52000e5 + (Py \wedge 2) / \dots \\
 & 0.52000e5 + ((Pz - 230) \wedge 2) / 0.52000e5 - 0.569e3 / 0.520e3) \wedge 2) \wedge \dots \\
 & (-0.1e1 / 0.2e1)) \wedge 2 * ((-4 * Px * Py \wedge 2 - 4 * Px * Pz \wedge 2 + 1840 * \dots \\
 & Px * Pz + 16000 * Px - 4 * Px \wedge 3) / ((Px \wedge 2) / 0.52000e5 + (Py \wedge 2) / \dots \\
 & 0.52000e5 + ((Pz - 230) \wedge 2) / 0.52000e5 - 0.569e3 / 0.520e3) \wedge 2 - \dots
 \end{aligned}$$

Licencia Creative Commons:

Atribución - No Comercial - Compartir Igual 3.0 Venezuela
(CC BY-NC-SA 3.0 VE)

$$\begin{aligned}
 & (2688000000 - 2 * Px \wedge 2 * Py \wedge 2 - 2 * Px \wedge 2 * Pz \wedge 2 + 920 * \dots \\
 & Px \wedge 2 * Pz - 2 * Py \wedge 2 * Pz \wedge 2 + 8000 * Py \wedge 2 - 3680000 * Pz + \dots \\
 & 8000 * Px \wedge 2 - 203600 * Pz \wedge 2 - Px \wedge 4 - Py \wedge 4 - Pz \wedge 4 + 920 * \dots \\
 & Py \wedge 2 * Pz + 920 * Pz \wedge 3) / ((Px \wedge 2) / 0.52000e5 + (Py \wedge 2) / \dots \\
 & 0.52000e5 + ((Pz - 230) \wedge 2) / 0.52000e5 - 0.569e3 / 0.520e3) \wedge 3 * \dots \\
 & Px / 0.13000e5)) / (0.1e1 + 0.16900e5 * (2688000000 - 2 * Px \wedge 2 * \dots \\
 & Py \wedge 2 - 2 * Px \wedge 2 * Pz \wedge 2 + 920 * Px \wedge 2 * Pz - 2 * Py \wedge 2 * \dots \\
 & Pz \wedge 2 + 8000 * Py \wedge 2 - 3680000 * Pz + 8000 * Px \wedge 2 - 203600 * \dots \\
 & Pz \wedge 2 - Px \wedge 4 - Py \wedge 4 - Pz \wedge 4 + 920 * Py \wedge 2 * Pz + 920 * \dots \\
 & Pz \wedge 3) / ((Px \wedge 2) / 0.52000e5 + (Py \wedge 2) / 0.52000e5 + \dots \\
 & ((Pz - 230) \wedge 2) / 0.52000e5 - 0.569e3 / 0.520e3) \wedge 2 / \dots \\
 & (0.2704000000e10 + (2688000000 - 2 * Px \wedge 2 * Py \wedge 2 - 2 * \dots \\
 & Px \wedge 2 * Pz \wedge 2 + 920 * Px \wedge 2 * Pz - 2 * Py \wedge 2 * Pz \wedge 2 + \dots \\
 & 8000 * Py \wedge 2 - 3680000 * Pz + 8000 * Px \wedge 2 - 203600 * Pz \wedge 2 - \dots \\
 & Px \wedge 4 - Py \wedge 4 - Pz \wedge 4 + 920 * Py \wedge 2 * Pz + 920 * Pz \wedge 3) / \dots \\
 & ((Px \wedge 2) / 0.52000e5 + (Py \wedge 2) / 0.52000e5 + ((Pz - 230) \wedge 2) / \dots \\
 & 0.52000e5 - 0.569e3 / 0.520e3) \wedge 2) / (0.200e3 + 0.6760000e7 * \dots \\
 & (0.2704000000e10 + (2688000000 - 2 * Px \wedge 2 * Py \wedge 2 - 2 * Px \wedge 2 * \dots \\
 & Pz \wedge 2 + 920 * Px \wedge 2 * Pz - 2 * Py \wedge 2 * Pz \wedge 2 + 8000 * Py \wedge 2 - \dots \\
 & 3680000 * Pz + 8000 * Px \wedge 2 - 203600 * Pz \wedge 2 - Px \wedge 4 - Py \wedge 4 - \dots \\
 & Pz \wedge 4 + 920 * Py \wedge 2 * Pz + 920 * Pz \wedge 3) / ((Px \wedge 2) / 0.52000e5 + \dots \\
 & (Py \wedge 2) / 0.52000e5 + ((Pz - 230) \wedge 2) / 0.52000e5 - 0.569e3 / \dots \\
 & 0.520e3) \wedge 2) \wedge (-0.1e1 / 0.2e1)) \wedge 2);
 \end{aligned}$$

$$\begin{aligned}
 \text{JinvGeomA}(2,2) &= -((Pz - 230) * (Px \wedge 2 + Py \wedge 2) \wedge (-0.3e1 / 0.2e1) * \dots \\
 & Py / (1 + (Pz - 230) \wedge 2 / (Px \wedge 2 + Py \wedge 2))) - (0.65e2 * ((2688000000 - \dots \\
 & 2 * Px \wedge 2 * Py \wedge 2 - 2 * Px \wedge 2 * Pz \wedge 2 + 920 * Px \wedge 2 * Pz - 2 * \dots \\
 & Py \wedge 2 * Pz \wedge 2 + 8000 * Py \wedge 2 - 3680000 * Pz + 8000 * Px \wedge 2 - 203600 * \dots \\
 & Pz \wedge 2 - Px \wedge 4 - Py \wedge 4 - Pz \wedge 4 + 920 * Py \wedge 2 * Pz + 920 * Pz \wedge 3) \wedge \dots \\
 & (-0.1e1 / 0.2e1)) / ((Px \wedge 2) / 0.52000e5 + (Py \wedge 2) / 0.52000e5 + ((Pz - 230) \wedge 2) / \\
 & 0.52000e5 - 0.569e3 / 0.520e3) * (0.2704000000e10 + (2688000000 - 2 * Px \wedge 2 * Py \wedge 2 - 2 \\
 & * Px \wedge 2 * Pz \wedge 2 + 920 * Px \wedge 2 * Pz - 2 * Py \wedge 2 * Pz \wedge 2 + 8000 * Py \wedge 2 - 3680000 * Pz \\
 & + 8000 * Px \wedge 2 - 203600 * Pz \wedge 2 - Px \wedge 4 - Py \wedge 4 - Pz \wedge 4 + 920 * Py \wedge 2 * Pz + 920 * \\
 & Pz \wedge 3) / ((Px \wedge 2) / 0.52000e5 + (Py \wedge 2) / 0.52000e5 + ((Pz - 230) \wedge 2) / 0.52000e5 - \\
 & 0.569e3 / 0.520e3) \wedge 2) \wedge (-0.1e1 / 0.2e1) / (0.200e3 + 0.6760000e7 * (0.2704000000e10 + \\
 & (2688000000 - 2 * Px \wedge 2 * Py \wedge 2 - 2 * Px \wedge 2 * Pz \wedge 2 + 920 * Px \wedge 2 * Pz - 2 * Py \wedge 2 * \\
 & Pz \wedge 2 + 8000 * Py \wedge 2 - 3680000 * Pz + 8000 * Px \wedge 2 - 203600 * Pz \wedge 2 - Px \wedge 4 - Py \wedge 4 \\
 & - Pz \wedge 4 + 920 * Py \wedge 2 * Pz + 920 * Pz \wedge 3) / ((Px \wedge 2) / 0.52000e5 + (Py \wedge 2) / \\
 & 0.52000e5 + ((Pz - 230) \wedge 2) / 0.52000e5 - 0.569e3 / 0.520e3) \wedge 2) \wedge (-0.1e1 / 0.2e1)) * \\
 & (-4 * Px \wedge 2 * Py - 4 * Py * Pz \wedge 2 + 16000 * Py - 4 * Py \wedge 3 + 1840 * Py * Pz) - \\
 & \text{sqrt}((2688000000 - 2 * Px \wedge 2 * Py \wedge 2 - 2 * Px \wedge 2 * Pz \wedge 2 + 920 * Px \wedge 2 * Pz - 2 * Py \\
 & \wedge 2 * Pz \wedge 2 + 8000 * Py \wedge 2 - 3680000 * Pz + 8000 * Px \wedge 2 - 203600 * Pz \wedge 2 - Px \wedge 4 -
 \end{aligned}$$

$$\begin{aligned}
 & (-4 * Px \wedge 2 * Py - 4 * Py * Pz \wedge 2 + 16000 * Py - 4 * Py \wedge 3 + 1840 * Py * Pz) - \\
 & \text{sqrt}((2688000000 - 2 * Px \wedge 2 * Py \wedge 2 - 2 * Px \wedge 2 * Pz \wedge 2 + 920 * Px \wedge 2 * Pz - 2 * Py \\
 & \wedge 2 * Pz \wedge 2 + 8000 * Py \wedge 2 - 3680000 * Pz + 8000 * Px \wedge 2 - 203600 * Pz \wedge 2 - Px \wedge 4 - \\
 & Py \wedge 4 - Pz \wedge 4 + 920 * Py \wedge 2 * Pz + 920 * Pz \wedge 3)) / ((Px \wedge 2) / 0.52000e5 + (Py \wedge 2) / \\
 & 0.52000e5 + ((Pz - 230) \wedge 2) / 0.52000e5 - 0.569e3 / 0.520e3) \wedge 2 * (0.2704000000e10 + \\
 & (2688000000 - 2 * Px \wedge 2 * Py \wedge 2 - 2 * Px \wedge 2 * Pz \wedge 2 + 920 * Px \wedge 2 * Pz - 2 * Py \wedge 2 * \\
 & Pz \wedge 2 + 8000 * Py \wedge 2 - 3680000 * Pz + 8000 * Px \wedge 2 - 203600 * Pz \wedge 2 - Px \wedge 4 - Py \wedge 4 \\
 & - Pz \wedge 4 + 920 * Py \wedge 2 * Pz + 920 * Pz \wedge 3) / ((Px \wedge 2) / 0.52000e5 + (Py \wedge 2) / \\
 & 0.52000e5 + ((Pz - 230) \wedge 2) / 0.52000e5 - 0.569e3 / 0.520e3) \wedge 2 * (-0.1e1 / 0.2e1) / \\
 & (0.200e3 + 0.6760000e7 * (0.2704000000e10 + (2688000000 - 2 * Px \wedge 2 * Py \wedge 2 - 2 * Px \wedge 2 \\
 & * Pz \wedge 2 + 920 * Px \wedge 2 * Pz - 2 * Py \wedge 2 * Pz \wedge 2 + 8000 * Py \wedge 2 - 3680000 * Pz
 \end{aligned}$$

WWW.BDIGITAL.ULA.VE

Anexo I

Programa en MATLAB® para el cálculo de las tres trayectorias, utilizando la cinemática inversa a partir de las matrices homogéneas.

WWW.BDIGITAL.ULA.VE

Licencia Creative Commons:
Atribución - No Comercial - Compartir Igual 3.0 Venezuela
(CC BY-NC-SA 3.0 VE)

```

close all;
clear all;
clc;

% Cinemática inversa calculada a partir de las matrices homogeneas

% -----
% Primera trayectoria, una línea recta a lo largo del eje X, y Z ctte.
% -----

Px = [100:10:200 ]',
Py = [0 0 0 0 0 0 0 0 0 0 0]',
Pz = [130 130 130 130 130 130 130 130 130 130 130]',
T = [0:1:10],

for i = 1:11

theta1(i) = atan(Py(i)/Px(i)),
a(i) = (cos(theta1(i))*Px(i)+sin(theta1(i))*Py(i)),
b(i) = (Pz(i)-230),
c(i) = (23100+a(i)^2+b(i)^2)/400, % (23100+a*a+b*b)/400;

theta2a(i) = 2*atan((b(i)+(sqrt(b(i)^2-c(i)^2+a(i)^2)))/(a(i)+c(i))),
theta2b(i) = 2*atan((b(i)-(sqrt(b(i)^2-c(i)^2+a(i)^2)))/(a(i)+c(i))),

theta3a(i) = atan((b(i)*cos(theta2a(i))-a(i)*sin(theta2a(i)))/...
(a(i)*cos(theta2a(i))+b(i)*sin(theta2a(i))-200)),
theta3b(i) = atan((b(i)*cos(theta2b(i))-a(i)*sin(theta2b(i)))/...
(a(i)*cos(theta2b(i))+b(i)*sin(theta2b(i))-200)),
end

[[theta1]' [theta2a]' [theta2b]' [theta3a]' [theta3b]'],

subplot (1,2,1), plot(Px,Py), title('Trayectoria (rad)'), xlabel('X (mm)'), ...
ylabel('Y (mm)'),

```

```

subplot (1,2,2), plot(T,theta1,'-r'), title ('Ángulos de las articulaciones'),...
    hold all, plot(T,theta2a,'-.k'), hold all, plot(T,theta3a,':b'),...
    hold all, plot(T,theta2b,'-ok'), hold all, plot(T,theta3b,':ob'),...
    legend('theta1','theta2a','theta3a','theta2b','theta3b',-1),...
    xlabel('Número de pasos'), ylabel ('Ángulos de las articulaciones (rad)'),

% -----
% Segunda trayectoria, una línea recta a lo largo del eje Y, con Z ccte.
% -----

Px = [50 50 50 50 50 50 50 50 50 50 50]',
Py = [0:20:200]',
Pz = [130 130 130 130 130 130 130 130 130 130 130]',
T = [0:1:10],

for i = 1:11

theta1(i) = atan(Py(i)/Px(i)),
a(i) = (cos(theta1(i))*Px(i)+sin(theta1(i))*Py(i)),
b(i) = (Pz(i)-230),
c(i) = (23100+a(i)^2+b(i)^2)/400, % (23100+a*a+b*b)/400;

theta2a(i) = 2*atan((b(i)+(sqrt(b(i)^2-c(i)^2+a(i)^2)))/(a(i)+c(i))),
theta2b(i) = 2*atan((b(i)-(sqrt(b(i)^2-c(i)^2+a(i)^2)))/(a(i)+c(i))),

theta3a(i) = atan((b(i)*cos(theta2a(i))-a(i)*sin(theta2a(i)))/...
    (a(i)*cos(theta2a(i))+b(i)*sin(theta2a(i))-200)),
theta3b(i) = atan((b(i)*cos(theta2b(i))-a(i)*sin(theta2b(i)))/...
    (a(i)*cos(theta2b(i))+b(i)*sin(theta2b(i))-200)),
end

[[theta1]' [theta2a]' [theta2b]' [theta3a]' [theta3b]'],

figure,
subplot (1,2,1), plot(Px,Py), title('Trayectoria'), xlabel('X (mm)'), ylabel('Y (mm)'),

```

Licencia Creative Commons:

Atribución - No Comercial - Compartir Igual 3.0 Venezuela
(CC BY-NC-SA 3.0 VE)

```

subplot (1,2,2), plot(T,theta1,'-r'), title('Ángulos de las articulaciones'), ...
    hold all, plot(T,theta2a,'-.k'), hold all, plot(T,theta3a,':b'), ...
    hold all, plot(T,theta2b,'-ok'), hold all, plot(T,theta3b,':ob'),...
    legend('theta1','theta2a','theta3a','theta2b','theta3b',-1),...
    xlabel('Número de pasos'), ylabel('Ángulos de las articulaciones (rad)'),

% -----
% Tercera trayectoria, una línea recta en el espacio.
% -----
% La ecuación de la recta viene dada por:
%
% (100-x) = (150-y) = (200-z)
% -----

Px = [100:5:150],
Py = Px,
Pz = Py + 50,
T = [0:1:10],

for i = 1:11

theta1(i) = atan(Py(i)/Px(i)),

a(i) = (cos(theta1(i))*Px(i)+sin(theta1(i))*Py(i)),
b(i) = (Pz(i)-230),
c(i) = (23100+a(i)^2+b(i)^2)/400, % (23100+a*a+b*b)/400;

theta2a(i) = 2*atan((b(i)+(sqrt(b(i)^2-c(i)^2+a(i)^2)))/(a(i)+c(i))),
theta2b(i) = 2*atan((b(i)-(sqrt(b(i)^2-c(i)^2+a(i)^2)))/(a(i)+c(i))),

theta3a(i) = atan((b(i)*cos(theta2a(i))-a(i)*sin(theta2a(i)))/...
    (a(i)*cos(theta2a(i))+b(i)*sin(theta2a(i))-200)),
theta3b(i) = atan((b(i)*cos(theta2b(i))-a(i)*sin(theta2b(i)))/...
    (a(i)*cos(theta2b(i))+b(i)*sin(theta2b(i))-200)),
end

```

```

[[theta1]' [theta2a]' [theta2b]' [theta3a]' [theta3b]'],

figure,
subplot (2,3,1), plot(Px,Py), title('Trayectoria'), grid on, ...
    axis ([0, 300, 0, 300]), xlabel('X (mm)'), ylabel('Y (mm)'),
subplot (2,3,2), plot(Px,Pz), title('Trayectoria'), grid on, ...
    axis ([0, 300, 0, 300,]), xlabel('X (mm)'), ylabel('Z (mm)'),
subplot (2,3,3), plot(Py,Pz), title('Trayectoria'), grid on, ...
    axis ([0, 300, 0, 300,]), xlabel('Y (mm)'), ylabel('Z (mm)'),
subplot (2,3,4), plot3(Px,Py,Pz), title('Trayectoria'), grid on, ...
    axis ([0, 300, 0, 300, 0, 560]), xlabel('X (mm)'), ylabel('Y (mm)'), ...
    zlabel('Z (mm)'),
subplot (2,3,5:6), plot(T,theta1,'-r'), title('Ángulos de las articulaciones'), ...
    hold all, plot(T,theta2a,'-.k'), hold all, plot(T,theta3a,':b'), ...
    hold all, plot(T,theta2b,'-ok'), hold all, plot(T,theta3b,':ob'),...
    legend('theta1', 'theta2a', 'theta3a', 'theta2b', 'theta3b',-1),...
    xlabel('Número de pasos'), ylabel('Ángulos de las articulaciones (rad)'),

```

WWW.BDIGITAL.ULA.VE

Anexo J

Programa en MATLAB ® para el cálculo de las tres trayectorias, utilizando la cinemática inversa calculada geoméricamente.

WWW.BDIGITAL.ULA.VE

Licencia Creative Commons:
Atribución - No Comercial - Compartir Igual 3.0 Venezuela
(CC BY-NC-SA 3.0 VE)

```

close all;
clear all;
clc;

% Cinemática inversa calculada geoméricamente

% -----
% Primera trayectoria, una línea recta a lo largo del eje X, y Z ccte.
% -----

Px = [100:10:200 ]',
Py = [0 0 0 0 0 0 0 0 0 0 0]',
Pz = [130 130 130 130 130 130 130 130 130 130 130]',
T = [0:1:10],

for i = 1:11

D(i) = (Px(i)^2+Py(i)^2+(Pz(i)-230)^2-200^2-130^2)/(2*200*130),
theta1(i) = atan(Py(i)/Px(i)),
theta3Up(i) = atan(sqrt(1-D(i)^2)/D(i)),
theta3Down(i) = atan(-sqrt(1-D(i)^2)/D(i)),
theta2a(i) = atan((Pz(i)-230)/sqrt(Px(i)^2+Py(i)^2))-atan((130*sin(theta3Up(i))/...
(200+130*cos(theta3Up(i))))),
theta2b(i) = atan((Pz(i)-230)/sqrt(Px(i)^2+Py(i)^2))-atan((130*sin(theta3Down(i))/...
(200+130*cos(theta3Down(i))))),

end

[[theta1]' [theta2a]' [theta2b]' [theta3Up]' [theta3Down]'],

subplot (1,2,1), plot(Px,Py), title('Trayectoria (rad)'), xlabel ('X (mm)'),...
ylabel('Y (mm)'),
subplot (1,2,2), plot(T,theta1,'-r'), title ('Ángulos de las articulaciones'),...
hold all, plot(T,theta2a,'-.k'), hold all, plot(T,theta3Up,':b'),...
hold all, plot(T,theta2b,'-ok'), hold all, plot(T,theta3Down,':ob'),...
legend('theta1','theta2a','theta3Up','theta2b','theta3Down',-1), ...

```

```

xlabel('Número de pasos'), ylabel('Ángulo de las articulaciones (rad)'),

% -----
% Segunda trayectoria, una línea recta a lo largo del eje Y, con Z cte.
% -----

Px = [50 50 50 50 50 50 50 50 50 50 50]',
Py = [0:20:200]',
Pz = [130 130 130 130 130 130 130 130 130 130 130]',
T = [0:1:10],

for i = 1:11

    D(i) = (Px(i)^2+Py(i)^2+(Pz(i)-230)^2-200^2-130^2)/(2*200*130),
    theta1(i) = atan(Py(i)/Px(i)),
    theta3Up(i) = atan(sqrt(1-D(i)^2)/D(i)),
    theta3Down(i) = atan(-sqrt(1-D(i)^2)/D(i)),
    theta2a(i) = atan((Pz(i)-230)/sqrt(Px(i)^2+Py(i)^2))-atan((130*sin(theta3Up(i)))/...
        (200+130*cos(theta3Up(i))))),
    theta2b(i) = atan((Pz(i)-230)/sqrt(Px(i)^2+Py(i)^2))-atan((130*sin(theta3Down(i)))/...
        (200+130*cos(theta3Down(i))))),

end

[[theta1]' [theta2a]' [theta2b]' [theta3Up]' [theta3Down]'],

figure,
subplot (1,2,1), plot(Px,Py), title('Trayectoria'), xlabel('X (mm)'), ylabel('Y (mm)'),
subplot (1,2,2), plot(T,theta1,'-r'), title('Ángulos de las articulaciones'), ...
    hold all, plot(T,theta2a,'-.k'), hold all, plot(T,theta3Up,':b'), ...
    hold all, plot(T,theta2b,'-ok'), hold all, plot(T,theta3Down,':ob'),...
    legend('theta1','theta2a','theta3Up','theta2b','theta3Down',-1), ...
    xlabel('Número de pasos'), ylabel('Ángulo de las articulaciones (rad)'),

% -----
% Tercera trayectoria, una línea recta en el espacio.

```

```

% -----
% La ecuación de la recta viene dada por:
%
% (100-x) = (150-y) = (200-z)
% -----

Px = [100:5:150],
Py = Px,
Pz = Py + 50,
T = [0:1:10],

for i = 1:11

    D(i) = (Px(i)^2+Py(i)^2+(Pz(i)-230)^2-200^2-130^2)/(2*200*130),
    theta1(i) = atan(Py(i)/Px(i)),
    theta3Up(i) = atan(sqrt(1-D(i)^2)/D(i)),
    theta3Down(i) = atan(-sqrt(1-D(i)^2)/D(i)),
    theta2a(i) = atan((Pz(i)-230)/sqrt(Px(i)^2+Py(i)^2))-atan((130*sin(theta3Up(i)))/...
        (200+130*cos(theta3Up(i))))),
    theta2b(i) = atan((Pz(i)-230)/sqrt(Px(i)^2+Py(i)^2))-atan((130*sin(theta3Down(i)))/...
        (200+130*cos(theta3Down(i))))),

end

[[theta1]' [theta2a]' [theta2b]' [theta3Up]' [theta3Down]'],

figure,
subplot (2,3,1), plot(Px,Py), title('Trayectoria'), grid on, ...
    axis ([0, 300, 0, 300]), xlabel('X (mm)'), ylabel('Y (mm)'),
subplot (2,3,2), plot(Px,Pz), title('Trayectoria'), grid on, ...
    axis ([0, 300, 0, 300,]), xlabel('X (mm)'), ylabel('Z (mm)'),
subplot (2,3,3), plot(Py,Pz), title('Trayectoria'), grid on, ...
    axis ([0, 300, 0, 300]), xlabel('Y (mm)'), ylabel('Z (mm)'),
subplot (2,3,4), plot3(Px,Py,Pz), title('Trayectoria'), grid on, ...
    axis ([0, 300, 0, 300, 0, 560]), xlabel('X (mm)'), ylabel('Y (mm)'), zlabel('Z (mm)'),
subplot (2,3,5:6), plot(T,theta1,'-r'), title('Ángulos de las articulaciones'), ...

```

```

hold all, plot(T,theta2a,'-.k'), hold all, plot(T,theta3Up,':b'), ...
hold all, plot(T,theta2b,'-ok'), hold all, plot(T,theta3Down,':ob'),...
legend('theta1','theta2a','theta3Up','theta2b','theta3Down',-1),...
xlabel('Número de pasos'), ylabel('Ángulo de las articulaciones (rad)'),

% -----
% Cuarta trayectoria, una línea recta a lo largo del eje Y, con Z ctte.
% -----

Px = [150 150 150 150 150 150 150 150 150 150 150]',
Py = [0:20:200]',
Pz = [130 130 130 130 130 130 130 130 130 130 130]',
T = [0:1:10],

for i = 1:11

    D(i) = (Px(i)^2+Py(i)^2+(Pz(i)-230)^2-200^2-130^2)/(2*200*130),
    theta1(i) = atan(Py(i)/Px(i)),
    theta3Up(i) = atan(sqrt(1-D(i)^2)/D(i)),
    theta3Down(i) = atan(-sqrt(1-D(i)^2)/D(i)),
    theta2a(i) = atan((Pz(i)-230)/sqrt(Px(i)^2+Py(i)^2))-atan((130*sin(theta3Up(i)))/...
        (200+130*cos(theta3Up(i))))),
    theta2b(i) = atan((Pz(i)-230)/sqrt(Px(i)^2+Py(i)^2))-atan((130*sin(theta3Down(i)))/...
        (200+130*cos(theta3Down(i))))),

end

[[theta1]' [theta2a]' [theta2b]' [theta3Up]' [theta3Down]'],

figure,
subplot (1,2,1), plot(Px,Py), title('Trayectoria'), xlabel('X (mm)'), ylabel('Y (mm)'),
subplot (1,2,2), plot(T,theta1,'-r'), title('Ángulos de las articulaciones'), ...
    hold all, plot(T,theta2a,'-.k'), hold all, plot(T,theta3Up,':b'), ...
    hold all, plot(T,theta2b,'-ok'), hold all, plot(T,theta3Down,':ob'),...
    legend('theta1','theta2a','theta3Up','theta2b','theta3Down',-1), ...
    xlabel('Número de pasos'), ylabel('Ángulo de las articulaciones (rad)'),

```

Anexo K

Programa en MATLAB ® utilizando “Robotic Toolbox for MATLAB” para el cálculo de trayectorias y torques aplicados a las articulaciones.

WWW.BDIGITAL.ULA.VE

Licencia Creative Commons:
Atribución - No Comercial - Compartir Igual 3.0 Venezuela
(CC BY-NC-SA 3.0 VE)

```

close all;
clear all;
clc;

% Luis José Andueza Castro
% Especialización y Maestría en Ingeniería de Control y Automatización
% Escuela de Ingeniería de Sistemas
% Universidad de Los Andes
%
%
% Programa para el cálculo de TRAYECTORIAS
%
%
% Los Parámetro de Denavit - Hartenberg (según Barrientos et Al.)
% que se tienen son:
%
% -----
% | Articulación | Theta | d | a | Alfa |
% -----
% | 1 | Theta 1 | 230 | 0 | 90 |
% -----
% | 2 | Theta 2 | 0 | 200 | 0 |
% -----
% | 3 | Theta 3 | 0 | 130 | 0 |
% -----
%
% Por lo que los parámetros Denavit - Hartenberg para el caso en estudio
% quedan como:
%
% -----
% | Articulación | Alfa | a | Theta | d | sigma |
% -----
% | 1 | pi/2 | 0 | 0 | 230 | 0 |
% -----
% | 2 | 0 | 200 | 0 | 0 | 0 |
% -----

```

```

% | 3 | 0 | 130 | 0 | 0 | 0 |
% -----
%
% Es de notar que el TOOLBOX utiliza radianes en vez de grados.
%
% Una vez obtenidos los parametros de Denavit - Hartenberg, se procede
% a definir los eslabones, donde cada eslabón estará definido con la letra
% L (link), donde L{1} es para el eslabón 1, L{2} para el eslabón 2 y así
% sucesivamente:

clear L;

% L{#} = número del eslabón y tipo
% L.m = masa del eslabón
% L.r = centro de masa del eslabón
% L.I = Tensores de inercia
% L.Jm = Inercia del motor que acciona el elemento
% L.G = Coeficiente de reducción del actuador del elemento
% L.B = Rozamiento viscoso del actuador del elemento
% L.Tc = Rozamiento de Coulomb en la dirección positiva y negativa del
% actuador del elemento

L{1} = link([pi/2, 0, 0, 230, 0]);
L{1}.m = 0.583;
L{1}.r = [0, -82, 0];
L{1}.I = [2422.309, 0,0; 4.285, 1587.979, 0; -0.2e-2, -0.2e-2, 1474.055];L{1}.Jm = 0;
L{1}.G = 1;
L{1}.B = 0;
L{1}.Tc = [0, 0];

L{2} = link ([0, 200, 0, 0, 0]);
L{2}.m = 0.473;
L{2}.r = [-86.78, 0, 13];
L{2}.I = [1059.248, 0,0; 1.678, 2089.822, 0; -99.320, 43.420, 1594.025];L{2}.Jm = 0;
L{2}.G = 1;
L{2}.B = 0;

```

```
L{2}.Tc = [0, 0];
```

```
L{3} = link ([0, 130, 0, 0, 0]);
```

```
L{3}.m = 0.153;
```

```
L{3}.r = [-78.8, 0, 0];
```

```
L{3}.I = [282.163, 0,0; 0, 284.792, 0; 0, 0, 262.959];L{3}.Jm = 0;
```

```
L{3}.G = 1;
```

```
L{3}.B = 0;
```

```
L{3}.Tc = [0, 0];
```

```
% Una vez definidos los eslabones del robot, se define el robot en sí.
```

```
Qz = [0 0 0]; % angulos todos en cero (0), posicion en L
```

```
BrazoRobotico = robot (L),
```

```
BrazoRobotico.name = 'Manipulador',
```

```
BrazoRobotico.gravity = [0, 0, 9810],
```

```
% BrazoRobotico.base = transl([0,0,0]),
```

```
% plot(BrazoRobotico, Qz),
```

```
% drivebot(BrazoRobotico),
```

```
% Se crearan diversas posiciones asignando valores de angulos para las  
% articulaciones rotativas, para la cinematica directa.
```

```
% Qz, la primera posición será en posición inicial con todas articulaciones  
% en cero grados.
```

```
Qz,
```

```
% Q1 la posición final para la primera trayectoria
```

```
Q1f = [0, 0.1505, 1.4377], % recordando que los grados tienen que ser radianes
```

```
% Q2 segunda trayectoria
```

```
Q2i = [0, -0.4571, 0.5475],
```

```
Q2f = [1.3258 0.1492 1.4861],
```

```
% Q3 Tercera trayectoria
```

```
q3i = [0.7854, 0.1902, 0.9440],
```

```
q3f = [0.7854, 0.4944, 1.3576],
```

```
%Una vez que se tienen las posiciones, se procede a probar la cinemática
%directa.
```

```
CdirectaQz = fkine(BrazoRobotico, Qz), % Cinemática directa para Qz
```

```
CdirectaQ1 = fkine(BrazoRobotico, Q1f), % Cinemática directa para Q1
```

```
CdirectaQ2 = fkine(BrazoRobotico, Q2f), % Cinemática directa para Q2
```

```
CdirectaQ3 = fkine(BrazoRobotico, Q3f), % Cinemática directa para Q3
```

```
t = [0:1:10];
```

```
% cálculo de trayectorias jtraj
```

```
qz = jtraj(Qz, Q1f, t);
```

```
q1 = jtraj(Q2i, Q2f, t);
```

```
q2 = jtraj(Q3i, Q3f, t);
```

```
figure
```

```
%subplot (1,2,1), plot(t,qz), title('Trayectoria (rad)'), xlabel('x (mm)'), ...
```

```
    % ylabel('y (mm)'),
```

```
subplot (1,3,1), plot(t,qz), title ('Trayectoria 1'),...
```

```
    legend('theta1','theta2','theta3',2),...
```

```
    xlabel('Número de pasos'), ylabel ('Ángulos (rad)'),
```

```
subplot (1,3,2), plot(t,q1), title ('Trayectoria 2'),...
```

```
    legend('theta1','theta2','theta3',2),...
```

```
    xlabel('Número de pasos'), ylabel ('Ángulos (rad)'),
```

```
subplot (1,3,3), plot(t,q2), title ('Trayectoria 3'),...
```

```
    legend('theta1','theta2','theta3',2),...
```

```
    xlabel('Número de pasos'), ylabel ('Ángulos (rad)'),
```

Anexo L

Programa en MATLAB ® para el cálculo de la Dinámica del robot manipulador propuesto.
Específicamente para obtener los torques aplicados en las articulaciones.

WWW.BDIGITAL.ULA.VE

Licencia Creative Commons:
Atribución - No Comercial - Compartir Igual 3.0 Venezuela
(CC BY-NC-SA 3.0 VE)

```
close all;
clear all;
clc;
```

```
% Luis José Andueza Castro
% Especialización y Maestría en Ingeniería de Control y Automatización
% Escuela de Ingeniería de Sistemas
% Universidad de Los Andes
```

```
%
%
```

```
% Programa para el cálculo de la Dinámica del Manipulador
```

```
%
%
```

```
% Los Parámetro de Denavit - Hartenberg (según Barrientos et Al.)
% que se tienen son:
```

```
%
```

```
% -----
% | Articulación | Theta | d | a | Alfa |
% -----
% | 1 | Theta 1 | 230 | 0 | 90 |
% -----
% | 2 | Theta 2 | 0 | 200 | 0 |
% -----
% | 3 | Theta 3 | 0 | 130 | 0 |
% -----
%
```

```
% Por lo que los parámetros Denavit - Hartenberg para el caso en estudio
% quedan como:
```

```
%
```

```
% -----
% | Articulación | Alfa | a | Theta | d | sigma |
% -----
% | 1 | pi/2 | 0 | 0 | 230 | 0 |
% -----
% | 2 | 0 | 200 | 0 | 0 | 0 |
% -----
% | 3 | 0 | 130 | 0 | 0 | 0 |
% -----
%
```

```
% Es de notar que el TOOLBOX utiliza radianes en vez de grados.
```

```
%
```

```
% Una vez obtenidos los parametros de Denavit - Hartenberg, se procede
% a definir los eslabones, donde cada eslabón estará definido con la letra
% L (link), donde L{1} es para el eslabón 1, L{2} para el eslabón 2 y así
% sucesivamente:
```

```
clear L;
```

```
% L{#} = número del eslabón y tipo
% L.m = masa del eslabón
% L.r = centro de masa del eslabón
% L.I = Tensores de inercia
% L.Jm = Inercia del motor que acciona el elemento
% L.G = Coeficiente de reducción del actuador del elemento
```

Licencia Creative Commons:

Atribución - No Comercial - Compartir Igual 3.0 Venezuela
(CC BY-NC-SA 3.0 VE)

```

% L.B = Rozamiento viscoso del actuador del elemento
% L.Tc = Rozamiento de Coulomb en la dirección positiva y negativa del
%       actuador del elemento

L{1} = link([pi/2, 0, 0, 230, 0]);
L{1}.m = 0.583;
L{1}.r = [-2.807, -82.111, 0];
L{1}.I = [2422.309, 0,0; 4.285, 1587.979, 0; -0.2e-2, -0.2e-2, 1474.055];
L{1}.Jm = 0;
L{1}.G = 1;
L{1}.B = 0;
L{1}.Tc = [0, 0];

L{2} = link ([0, 200, 0, 0, 0]);
L{2}.m = 0.473;
L{2}.r = [-86.769, -1.732, 13.092];
L{2}.I = [1059.248, 0,0; 1.678, 2089.822, 0; -99.320, 43.420, 1594.025];
L{2}.Jm = 0;
L{2}.G = 1;
L{2}.B = 0;
L{2}.Tc = [0, 0];

L{3} = link ([0, 130, 0, 0, 0]);
L{3}.m = 0.153;
L{3}.r = [-78.809, 0, 0];
L{3}.I = [282.163, 0,0; 0, 284.792, 0; 0, 0, 262.959];
L{3}.Jm = 0;
L{3}.G = 1;
L{3}.B = 0;
L{3}.Tc = [0, 0];

% Una vez definidos los eslabones del robot, se define el robot en sí.

Qz = [0 0 0]; % ángulos todos en cero (0), posición en L

BrazoRobotico = robot (L),
BrazoRobotico.name = 'Manipulador',
BrazoRobotico.gravity = [0, 0, 9810],
% BrazoRobotico.base = transl([0,0,0]),

% plot(BrazoRobotico, Qz),
% drivebot(BrazoRobotico),

% Se crearan diversas posiciones asignando valores de ángulos para las
% articulaciones rotativas, para la cinemática directa.

% Qz, la primera posición será en posición inicial con todas articulaciones
% en cero grados.

Qz,

% Q1 la posición final para la primera trayectoria
Q1f = [0, 0.1505, 1.4377], % recordando que los grados tienen que ser radianes

```

```

% Q2 Segunda trayectoria
Q2i = [0, -0.4571, 0.5475],
Q2f = [1.3258 0.1492 1.4861],

% Q3 Tercera trayectoria
Q3i = [0.7854, 0.1902, 0.9440],
Q3f = [0.7854, 0.4944, 1.3576],

%Una vez que se tienen las posiciones, se procede a probar la cinematica
%directa.

cdirectaQz = fkine(BrazoRobotico, Qz), % Cinematica directa para Qz
cdirectaQ1 = fkine(BrazoRobotico, Q1f), % Cinematica directa para Q1
cdirectaQ2 = fkine(BrazoRobotico, Q2f), % Cinematica directa para Q2
cdirectaQ3 = fkine(BrazoRobotico, Q3f), % Cinematica directa para Q3

t = [0:1:100];

% calculo de trayectorias jtraj
[qz qdz qddz] = jtraj(Qz, Q1f, t);
[q01 qd01 qdd01] = jtraj(Q2i, Q2f, t);
[q02 qd02 qdd02] = jtraj(Q3i, Q3f, t);

grav = [0 0 9810];

% tauz = (rne(BrazoRobotico, qz, qdz, qddz, grav))/100;
% tau1 = (rne(BrazoRobotico, q1, qd1, qdd1, grav))/100;
% tau2 = (rne(BrazoRobotico, q2, qd2, qdd2, grav))/100;
tauz = (rne(BrazoRobotico, qz, qdz, qddz, grav))/10000;
tau1 = (rne(BrazoRobotico, q01, qd01, qdd01, grav))/10000;
tau2 = (rne(BrazoRobotico, q02, qd02, qdd02, grav))/10000;

% *****
% -----
% Dinamica del manipulador por el metodo de Newton-Euler
% Algoritmo hecho en Matlab directamente
% -----
% *****
for j = 1:length(t)
theta1 = qz(j,1);
theta2 = qz(j,2);
theta3 = qz(j,3);

qd1 = qdz(j,1);
qd2 = qdz(j,2);
qd3 = qdz(j,3);

qdd1 = qddz(j,1);
qdd2 = qddz(j,2);
qdd3 = qddz(j,3);

```

```

% Establecimiento de condiciones iniciales
% -----
% velocidades y aceleraciones de la base del robot

gz = 9810                % Gravedad mm/seg2

VelAngIni = [0 0 0]';    % Velocidad angular inicial
AceAngIni = [0 0 0]';    % Aceleracion angular inicial
VelLinIni = [0 0 0]';    % Velocidad lineal inicial
AceLinIni = [0 0 gz]';   % Aceleracion lineal inicial
zo         = [0 0 1]';   %

% Matriz de rotacion entre {s3} y {s4}
% -----

R34 = eye(3);

% Coordenadas de los eslabones con respecto al eslabon anterior
% -----

p1 = [0 230 0]';
p2 = [200 0 0]';
p3 = [130 0 0]';

% Coordenadas del Centro de Gravedad
% -----

s1 = [-2.87 -82.111 0]';
s2 = [-86.769 -1.732 13.092]';
s3 = [-78.809 0 0]';

% Matrices de inercia de los eslabones respecto a sus centros de gravedad
% -----

I1 = [2422.309 0 0 ; 4.285 1587.979 0 ; -0.002 -0.002 1474.055];
I2 = [1059.248 0 0 ; 1.678 2089.822 0 ; -99.320 43.42 1594.025];
I3 = [282.163 0 0 ; 0 284.792 0 ; 0 0 262.959];

% Obtencion de las matrices de rotacion
% -----

R01 = [cos(theta1) 0 sin(theta1) ; sin(theta1) 0 -cos(theta1) ; 0 1 0];
R12 = [cos(theta2) -sin(theta2) 0 ; sin(theta2) cos(theta2) 0 ; 0 0 1];
R23 = [cos(theta3) -sin(theta3) 0 ; sin(theta3) cos(theta3) 0 ; 0 0 1];

% Evaluacion de las velocidades angulares del sistema {Si}
% -----

```

```

VelAng1 = R01' * (VelAngIni + Zo * qd1);
VelAng2 = R12' * (VelAng1 + Zo * qd2);
VelAng3 = R23' * (VelAng2 + Zo * qd3);

```

```

% Evaluaciones de las aceleraciones angulares del sistema {Si}
% -----

```

```

AceAng1 = R01' * (AceAngIni + Zo * qdd1) + cross(VelAngIni,Zo*qd1);
AceAng2 = R12' * (AceAng1 + Zo * qdd2) + cross(VelAng1,Zo*qd2);
AceAng3 = R23' * (AceAng2 + Zo * qdd3) + cross(VelAng2,Zo*qd3);

```

```

% Evaluacion de las aceleraciones lineales del sistema {Si}
% -----

```

```

AceLin1 = cross(AceAng1,P1) + cross(VelAng1, cross(VelAng1,P1)) + R01' * AceLinIni;
AceLin2 = cross(AceAng2,P2) + cross(VelAng2, cross(VelAng2,P2)) + R12' * AceLin1;
AceLin3 = cross(AceAng3,P3) + cross(VelAng3, cross(VelAng3,P3)) + R23' * AceLin2;

```

```

% Evaluacion de las aceleraciones lineales de los Centro de Gravedad
% -----

```

```

A1 = cross(AceAng1,S1) + cross(VelAng1, cross(VelAng1,S1)) + AceLin1;
A2 = cross(AceAng2,S2) + cross(VelAng2, cross(VelAng2,S2)) + AceLin2;
A3 = cross(AceAng3,S3) + cross(VelAng3, cross(VelAng3,S3)) + AceLin3;

```

```

% Evaluacion de las fuerzas sobre los eslabones
% -----

```

```
F4 = [0 0 0]';
```

```

m1 = 0.583;    % Masa eslabon 1 (Kg)
m2 = 0.473;    % Masa eslabon 2 (Kg)
m3 = 0.153;    % Masa eslabon 3 (Kg)

```

```

F3 = R34*F4 + m3*A3;
F2 = R23*F3 + m2*A2;
F1 = R12*F2 + m1*A1;

```

```

% Evaluacion de los pares ejercidos sobre los eslabones
% -----

```

```
n4 = [0 0 0]';
```

```

n3 = R34 * (n4 + cross((R34'*P3),F4)) + cross((P3+S3),m3*A3) + I3*AceAng3 +
cross(VelAng3, (I3*VelAng3));
n2 = R23 * (n3 + cross((R23'*P2),F3)) + cross((P2+S2),m2*A2) + I2*AceAng2 +
cross(VelAng2, (I2*VelAng2));
n1 = R12 * (n2 + cross((R12'*P1),F2)) + cross((P1+S1),m1*A1) + I1*AceAng1 +
cross(VelAng1, (I1*VelAng1));

```

```
% Evaluacion de los pares (Torques)
```

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

```
T1(j) = (n1' * R01' * Zo)/10000;
```

```
T2(j) = (n2' * R12' * Zo)/10000;
```

```
T3(j) = (n3' * R23' * Zo)/10000;
```

```
end
```

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

```
% Gráficos
```

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

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

```
% Trayectorias
```

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

```
% figure
```

```
% subplot (1,3,1), plot(t,qz), title ('Trayectoria 1'),...
```

```
% legend('theta1','theta2','theta3',2),...
```

```
% xlabel('Número de pasos'), ylabel ('Ángulos (rad)'),
```

```
% subplot (1,3,2), plot(t,q1), title ('Trayectoria 2'),...
```

```
% legend('theta1','theta2','theta3',2),...
```

```
% xlabel('Número de pasos'), ylabel ('Ángulos (rad)'),
```

```
% subplot (1,3,3), plot(t,q2), title ('Trayectoria 3'),...
```

```
% legend('theta1','theta2','theta3',2),...
```

```
% xlabel('Número de pasos'), ylabel ('Ángulos (rad)'),
```

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

```
% Grafica de Torque con Robotic Toolbox for Matlab
```

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

```
figure, title('Trayectoria 1 - Robotic Toolbox'),
```

```
subplot(1,3,1), plot(t, tauz(:,1)), ...
```

```
    title('Torque ejercido sobre la articulación 1'), xlabel('Número de pasos'), ...
```

```
    ylabel('Torque'), legend('Tau1',2),
```

```
subplot(1,3,2), plot(t, tauz(:,2)), ...
```

```
    title('Torque ejercido sobre la articulación 2'), xlabel('Número de pasos'), ...
```

```
    ylabel('Torque'), legend('Tau1',1),
```

```
subplot(1,3,3), plot(t, tauz(:,3)), ...
```

```
    title('Torque ejercido sobre la articulación 3'), xlabel('Número de pasos'), ...
```

```
    ylabel('Torque'), legend('Tau1',1),
```

```
% figure
```

```
% subplot(1,3,1), plot(t, tauz(:,1), t, tauz(:,2), t, tauz(:,3)), ...
```

```
%    title('Torque sobre las articulaciones, trayectoria 1'), xlabel('Número de pasos'), ...
```

```
%    ylabel('Torque'), legend('Tau1', 'Tau2', 'Tau3', 1),
```

```
% subplot(1,3,2), plot(t, tau1(:,1), t, tau1(:,2), t, tau1(:,3)), ...
```

```
%    title('Torque sobre las articulaciones, trayectoria 2'), xlabel('Número de pasos'), ...
```

Licencia Creative Commons:

Atribución - No Comercial - Compartir Igual 3.0 Venezuela
(CC BY-NC-SA 3.0 VE)

```
% ylabel('Torque'), legend('Tau1', 'Tau2', 'Tau3', 1),
%
% subplot(1,3,3), plot(t, tau2(:,1), t, tau2(:,2), t, tau2(:,3)), ...
% title('Torque sobre las articulaciones, trayectoria 3'), xlabel('Número de pasos'), ...
% ylabel('Torque'), legend('Tau1', 'Tau2', 'Tau3', 1),
```

```
% -----
% Gráfica de Torque con Newton - Euler
% -----
figure, title('Trayectoria 1 - Calculos Manuales'),
subplot(1,3,1), plot(t,T1), ...
    title('Torque ejercido sobre la articulación 1'), ...
    xlabel('Número de pasos'), ylabel('Torque (N.cm)'), ...

subplot(1,3,2), plot(t,T2), title('Torque ejercido sobre la articulación 2'), ...
    xlabel('Número de pasos'), ylabel('Torque (N.cm)'),

subplot(1,3,3), plot(t,T3), title('Torque ejercido sobre la articulación 3'), ...
    xlabel('Número de pasos'), ylabel('Torque (N.cm)'),
```

```
% -----
% Gráfica de comparación de los torques con el Toolbox y Manuales
% -----
```

```
figure, title('Trayectoria 1 - Robotic Toolbox'),
subplot(1,3,1), plot(t, tauz(:,1), t, T1, 'or'), ...
    title('Torque sobre la articulación 1'), xlabel('Número de pasos'), ...
    ylabel('Torque'), legend('Toolbox', 'Manual', 2),

subplot(1,3,2), plot(t, tauz(:,2), t, T2, 'or'), ...
    title('Torque ejercido sobre la articulación 2'), xlabel('Número de pasos'), ...
    ylabel('Torque'), legend('Toolbox', 'Manual', 1),

subplot(1,3,3), plot(t, tauz(:,3), t, T3, 'or'), ...
    title('Torque ejercido sobre la articulación 3'), xlabel('Número de pasos'), ...
    ylabel('Torque'), legend('Toolbox', 'Manual', 1),
```

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

```
% *****
% -----
% Dinamica del manipulador por el metodo de Newton-Euler
% Algoritmo hecho en Matlab directamente
% Segunda trayectoria
% -----
% *****
```

```
for j = 1:length(t)
theta1 = q01(j,1);
theta2 = q01(j,2);
```

```
theta3 = q01(j,3);
```

```
qd1 = qd01(j,1);
```

```
qd2 = qd01(j,2);
```

```
qd3 = qd01(j,3);
```

```
qdd1 = qdd01(j,1);
```

```
qdd2 = qdd01(j,2);
```

```
qdd3 = qdd01(j,3);
```

```
% Establecimiento de condiciones iniciales
```

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

```
% Velocidades y aceleraciones de la base del robot
```

```
gz = 9810
```

```
% Gravedad mm/seg2
```

```
VelAngIni = [0 0 0]'; % velocidad angular inicial
```

```
AceAngIni = [0 0 0]'; % Aceleracion angular inicial
```

```
VelLinIni = [0 0 0]'; % velocidad lineal inicial
```

```
AceLinIni = [0 0 gz]'; % Aceleracion lineal inicial
```

```
zo = [0 0 1]'; %
```

```
% Matriz de rotacion entre {S3} y {S4}
```

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

```
R34 = eye(3);
```

```
% Coordenadas de los eslabones con respecto al eslabon anterior
```

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

```
P1 = [0 230 0]';
```

```
P2 = [200 0 0]';
```

```
P3 = [130 0 0]';
```

```
% Coordenadas del Centro de Gravedad
```

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

```
s1 = [-2.87 -82.111 0]';
```

```
s2 = [-86.769 -1.732 13.092]';
```

```
s3 = [-78.809 0 0]';
```

```
% Matrices de inercia de los eslabones respecto a sus centros de gravedad
```

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

```
I1 = [2422.309 0 0 ; 4.285 1587.979 0 ; -0.002 -0.002 1474.055];
```

```
I2 = [1059.248 0 0 ; 1.678 2089.822 0 ; -99.320 43.42 1594.025];
```

```
I3 = [282.163 0 0 ; 0 284.792 0 ; 0 0 262.959];
```

% Obtencion de las matrices de rotacion

% -----

```
R01 = [cos(theta1) 0 sin(theta1) ; sin(theta1) 0 -cos(theta1) ; 0 1 0];
R12 = [cos(theta2) -sin(theta2) 0 ; sin(theta2) cos(theta2) 0 ; 0 0 1];
R23 = [cos(theta3) -sin(theta3) 0 ; sin(theta3) cos(theta3) 0 ; 0 0 1];
```

% Evaluacion de las velocidades angulares del sistema {Si}

% -----

```
VelAng1 = R01' * (VelAngIni + Zo * qd1);
VelAng2 = R12' * (VelAng1 + Zo * qd2);
VelAng3 = R23' * (VelAng2 + Zo * qd3);
```

% Evaluaciones de las aceleraciones angulares del sistema {Si}

% -----

```
AceAng1 = R01' * (AceAngIni + Zo * qdd1) + cross(VelAngIni,Zo*qd1);
AceAng2 = R12' * (AceAng1 + Zo * qdd2) + cross(VelAng1,Zo*qd2);
AceAng3 = R23' * (AceAng2 + Zo * qdd3) + cross(VelAng2,Zo*qd3);
```

% Evaluacion de las aceleraciones lineales del sistema {Si}

% -----

```
AceLin1 = cross(AceAng1,P1) + cross(VelAng1, cross(VelAng1,P1)) + R01' * AceLinIni;
AceLin2 = cross(AceAng2,P2) + cross(VelAng2, cross(VelAng2,P2)) + R12' * AceLin1;
AceLin3 = cross(AceAng3,P3) + cross(VelAng3, cross(VelAng3,P3)) + R23' * AceLin2;
```

% Evaluacion de las aceleraciones lineales de los Centro de Gravedad

% -----

```
A1 = cross(AceAng1,S1) + cross(VelAng1,cross(VelAng1,S1)) + AceLin1;
A2 = cross(AceAng2,S2) + cross(VelAng2,cross(VelAng2,S2)) + AceLin2;
A3 = cross(AceAng3,S3) + cross(VelAng3,cross(VelAng3,S3)) + AceLin3;
```

% Evaluacion de las fuerzas sobre los eslabones

% -----

```
F4 = [0 0 0]';
```

```
m1 = 0.583;    % Masa eslabon 1 (Kg)
m2 = 0.473;    % Masa eslabon 2 (Kg)
m3 = 0.153;    % Masa eslabon 3 (Kg)
```

```
F3 = R34*F4 + m3*A3;
F2 = R23*F3 + m2*A2;
F1 = R12*F2 + m1*A1;
```

```
% Evaluacion de los pares ejercidos sobre los eslabones
```

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

```
n4 = [0 0 0]';
```

```
n3 = R34 * (n4 + cross((R34'*P3),F4)) + cross((P3+S3),m3*A3) + I3*AceAng3 +  
cross(VelAng3,(I3*VelAng3));
```

```
n2 = R23 * (n3 + cross((R23'*P2),F3)) + cross((P2+S2),m2*A2) + I2*AceAng2 +  
cross(VelAng2,(I2*VelAng2));
```

```
n1 = R12 * (n2 + cross((R12'*P1),F2)) + cross((P1+S1),m1*A1) + I1*AceAng1 +  
cross(VelAng1,(I1*VelAng1));
```

```
% Evaluacion de los pares (Torques)
```

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

```
T1(j) = (n1' * R01' * Zo)/10000;
```

```
T2(j) = (n2' * R12' * Zo)/10000;
```

```
T3(j) = (n3' * R23' * Zo)/10000;
```

```
end
```

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

```
% Gráficos
```

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

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

```
% Grafica de Torque con Robotic Toolbox for Matlab
```

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

```
figure, title('Trayectoria 2 - Robotic Toolbox'),
```

```
subplot(1,3,1), plot(t, tau1(:,1)), ...
```

```
    title('Torque ejercido sobre la articulación 1'), xlabel('Número de pasos'), ...
```

```
    ylabel('Torque'), legend('Tau1',2),
```

```
subplot(1,3,2), plot(t, tau1(:,2)), ...
```

```
    title('Torque ejercido sobre la articulación 2'), xlabel('Número de pasos'), ...
```

```
    ylabel('Torque'), legend('Tau1',1),
```

```
subplot(1,3,3), plot(t, tau1(:,3)), ...
```

```
    title('Torque ejercido sobre la articulación 3'), xlabel('Número de pasos'), ...
```

```
    ylabel('Torque'), legend('Tau1',1),
```

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

```
% Gráfica de Torque con Newton - Euler
```

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

```
figure, title('Trayectoria 2 - Calculos Manuales'),
```

```
subplot(1,3,1), plot(t,T1), ...
```

```
    title('Torque ejercido sobre la articulación 1'), ...
```

```
    xlabel('Número de pasos'), ylabel('Torque (N.cm)'), ...
```

```
subplot(1,3,2), plot(t,T2), title('Torque ejercido sobre la articulación 2'), ...
```

Licencia Creative Commons:

Atribución - No Comercial - Compartir Igual 3.0 Venezuela
(CC BY-NC-SA 3.0 VE)

```

    xlabel('Número de pasos'), ylabel('Torque (N.cm)'),

subplot(1,3,3), plot(t,T3), title('Torque ejercido sobre la articulación 3'), ...
    xlabel('Número de pasos'), ylabel('Torque (N.cm)'),

% -----
% Gráfica de comparación de los torques con el Toolbox y Manuales
% -----

figure, title('Trayectoria 2 - Robotic Toolbox'),
subplot(1,3,1), plot(t, tau1(:,1), t, T1, 'or'), ...
    title('Torque sobre la articulación 1'), xlabel('Número de pasos'), ...
    ylabel('Torque'), legend('Toolbox', 'Manual', 2),

subplot(1,3,2), plot(t, tau1(:,2), t, T2, 'or'), ...
    title('Torque ejercido sobre la articulación 2'), xlabel('Número de pasos'), ...
    ylabel('Torque'), legend('Toolbox', 'Manual', 1),

subplot(1,3,3), plot(t, tau1(:,3), t, T3, 'or'), ...
    title('Torque ejercido sobre la articulación 3'), xlabel('Número de pasos'), ...
    ylabel('Torque'), legend('Toolbox', 'Manual', 1),

% -----
% -----
% -----
% *****
% -----
% Dinamica del manipulador por el metodo de Newton-Euler
% Algoritmo hecho en Matlab directamente
% Segunda trayectoria
% -----
% *****

for j = 1:length(t)
theta1 = q02(j,1);
theta2 = q02(j,2);
theta3 = q02(j,3);

qd1 = qd02(j,1);
qd2 = qd02(j,2);
qd3 = qd02(j,3);

qdd1 = qdd02(j,1);
qdd2 = qdd02(j,2);
qdd3 = qdd02(j,3);

% Establecimiento de condiciones iniciales
% -----
% Velocidades y aceleraciones de la base del robot

```

```

gz = 9810                % Gravedad mm/seg2

velAngIni = [0 0 0]';    % velocidad angular inicial
AceAngIni = [0 0 0]';    % Aceleracion angular inicial
velLinIni = [0 0 0]';    % velocidad lineal inicial
AceLinIni = [0 0 gz]';   % Aceleracion lineal inicial
zo         = [0 0 1]';   %

% Matriz de rotacion entre {S3} y {S4}
% -----

R34 = eye(3);

% Coordenadas de los eslabones con respecto al eslabon anterior
% -----

P1 = [0 230 0]';
P2 = [200 0 0]';
P3 = [130 0 0]';

% Coordenadas del Centro de Gravedad
% -----

S1 = [-2.87 -82.111 0]';
S2 = [-86.769 -1.732 13.092]';
S3 = [-78.809 0 0]';

% Matrices de inercia de los eslabones respecto a sus centros de gravedad
% -----

I1 = [2422.309 0 0 ; 4.285 1587.979 0 ; -0.002 -0.002 1474.055];
I2 = [1059.248 0 0 ; 1.678 2089.822 0 ; -99.320 43.42 1594.025];
I3 = [282.163 0 0 ; 0 284.792 0 ; 0 0 262.959];

% Obtencion de las matrices de rotacion
% -----

R01 = [cos(theta1) 0 sin(theta1) ; sin(theta1) 0 -cos(theta1) ; 0 1 0];
R12 = [cos(theta2) -sin(theta2) 0 ; sin(theta2) cos(theta2) 0 ; 0 0 1];
R23 = [cos(theta3) -sin(theta3) 0 ; sin(theta3) cos(theta3) 0 ; 0 0 1];

% Evaluacion de las velocidades angulares del sistema {Si}
% -----

velAng1 = R01' * (velAngIni + zo * qd1);
velAng2 = R12' * (velAng1 + zo * qd2);
velAng3 = R23' * (velAng2 + zo * qd3);

```

```
% Evaluaciones de las aceleraciones angulares del sistema {Si}
% -----
```

```
AceAng1 = R01' * (AceAngIni + Zo * qdd1) + cross(VelAngIni,Zo*qd1);
AceAng2 = R12' * (AceAng1 + Zo * qdd2) + cross(VelAng1,Zo*qd2);
AceAng3 = R23' * (AceAng2 + Zo * qdd3) + cross(VelAng2,Zo*qd3);
```

```
% Evaluacion de las aceleraciones lineales del sistema {Si}
% -----
```

```
AceLin1 = cross(AceAng1,P1) + cross(VelAng1, cross(VelAng1,P1)) + R01' * AceLinIni;
AceLin2 = cross(AceAng2,P2) + cross(VelAng2, cross(VelAng2,P2)) + R12' * AceLin1;
AceLin3 = cross(AceAng3,P3) + cross(VelAng3, cross(VelAng3,P3)) + R23' * AceLin2;
```

```
% Evaluacion de las aceleraciones lineales de los Centro de Gravedad
% -----
```

```
A1 = cross(AceAng1,S1) + cross(VelAng1,cross(VelAng1,S1)) + AceLin1;
A2 = cross(AceAng2,S2) + cross(VelAng2,cross(VelAng2,S2)) + AceLin2;
A3 = cross(AceAng3,S3) + cross(VelAng3,cross(VelAng3,S3)) + AceLin3;
```

```
% Evaluacion de las fuerzas sobre los eslabones
% -----
```

```
F4 = [0 0 0]';
```

```
m1 = 0.583;    % Masa eslabon 1 (Kg)
m2 = 0.473;    % Masa eslabon 2 (Kg)
m3 = 0.153;    % Masa eslabon 3 (Kg)
```

```
F3 = R34*F4 + m3*A3;
F2 = R23*F3 + m2*A2;
F1 = R12*F2 + m1*A1;
```

```
% Evaluacion de los pares ejercidos sobre los eslabones
% -----
```

```
n4 = [0 0 0]';
```

```
n3 = R34 * (n4 + cross((R34'*P3),F4)) + cross((P3+S3),m3*A3) + I3*AceAng3 +
cross(VelAng3,(I3*VelAng3));
n2 = R23 * (n3 + cross((R23'*P2),F3)) + cross((P2+S2),m2*A2) + I2*AceAng2 +
cross(VelAng2,(I2*VelAng2));
n1 = R12 * (n2 + cross((R12'*P1),F2)) + cross((P1+S1),m1*A1) + I1*AceAng1 +
cross(VelAng1,(I1*VelAng1));
```

```
% Evaluacion de los pares (Torques)
% -----
```

```

T1(j) = (n1' * R01' * Zo)/10000;
T2(j) = (n2' * R12' * Zo)/10000;
T3(j) = (n3' * R23' * Zo)/10000;

```

```
end
```

```

% -----
% Gráficos
% -----

% -----
% Grafica de Torque con Robotic Toolbox for Matlab
% -----

```

```

figure, title('Trayectoria 3 - Robotic Toolbox'),
subplot(1,3,1), plot(t, tau2(:,1)), ...
    title('Torque ejercido sobre la articulación 1'), xlabel('Número de pasos'), ...
    ylabel('Torque'), legend('Tau1',2),

subplot(1,3,2), plot(t, tau2(:,2)), ...
    title('Torque ejercido sobre la articulación 2'), xlabel('Número de pasos'), ...
    ylabel('Torque'), legend('Tau1',1),

subplot(1,3,3), plot(t, tau2(:,3)), ...
    title('Torque ejercido sobre la articulación 3'), xlabel('Número de pasos'), ...
    ylabel('Torque'), legend('Tau1',1),

```

```

% -----
% Gráfica de Torque con Newton - Euler
% -----

figure, title('Trayectoria 3 - Calculos Manuales'),
subplot(1,3,1), plot(t,T1), ...
    title('Torque ejercido sobre la articulación 1'), ...
    xlabel('Número de pasos'), ylabel('Torque (N.cm)'), ...

subplot(1,3,2), plot(t,T2), title('Torque ejercido sobre la articulación 2'), ...
    xlabel('Número de pasos'), ylabel('Torque (N.cm)'),

subplot(1,3,3), plot(t,T3), title('Torque ejercido sobre la articulación 3'), ...
    xlabel('Número de pasos'), ylabel('Torque (N.cm)'),

```

```

% -----
% Gráfica de comparación de los torques con el Toolbox y Manuales
% -----

```

```

figure, title('Trayectoria 2 - Robotic Toolbox'),
subplot(1,3,1), plot(t, tau2(:,1), t, T1, 'or'), ...
    title('Torque sobre la articulación 1'), xlabel('Número de pasos'), ...
    ylabel('Torque'), legend('Toolbox', 'Manual', 2),

subplot(1,3,2), plot(t, tau2(:,2), t, T2, 'or'), ...

```

```
title('Torque ejercido sobre la articulación 2'), xlabel('Número de pasos'), ...  
ylabel('Torque'), legend('Toolbox', 'Manual',1),  
  
subplot(1,3,3). plot(t, tau2(:,3), t, T3, 'or'), ...  
title('Torque ejercido sobre la articulación 3'), xlabel('Número de pasos'), ...  
ylabel('Torque'), legend('Toolbox', 'Manual',1),
```

WWW.BDIGITAL.ULA.VE

Anexo M

Centros de Gravedad y Matrices de Inercias

WWW.BDIGITAL.ULA.VE

Licencia Creative Commons:
Atribución - No Comercial - Compartir Igual 3.0 Venezuela
(CC BY-NC-SA 3.0 VE)

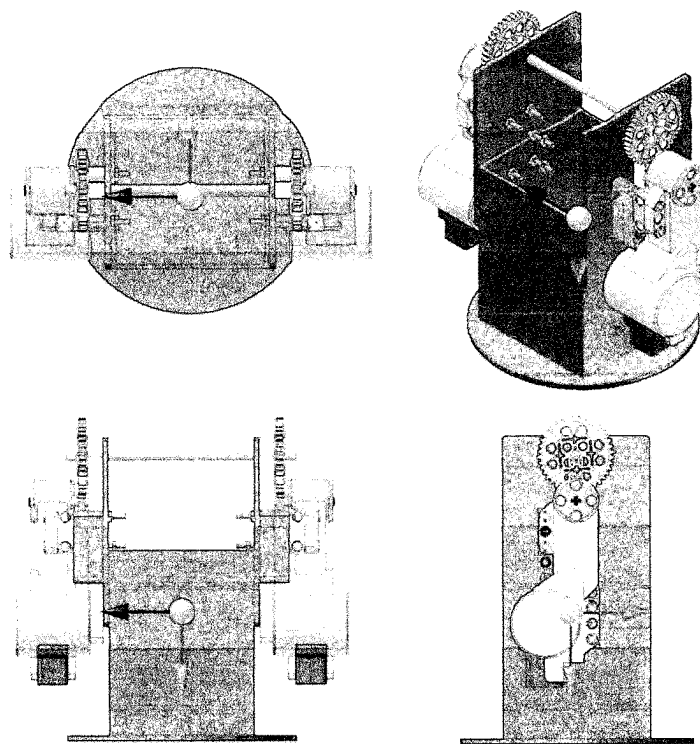


Figura 89 Centro de gravedad del eslabón 1

WWW.BDIGITAL.ULA.VE

General Properties

<input type="checkbox"/> Include Cosmetic Welds	<input type="checkbox"/> Include QTY Overrides
Mass : 0,584 kg (Relative)	Center of Gravity*
Area : 203459,960 mm ²	X : -2,807 mm (Relative)
Volume : 323470,856 mm ³	Y : 82,111 mm (Relative)
	Z : -0,000 mm (Relative)

Inertial Properties*

Principal Global Center Of Gravity

Mass Moments

Ixx : 2422,309 kg m	Calculated using negative integral.	
Ixy : 4,285 kg mm ²	Iyy : 1587,979 kg m	
Ixz : -0,002 kg mm ²	Iyz : -0,002 kg mm ²	Izz : 1474,055 kg m

*Values do not reflect user-overridden mass or volume

Figura 90 Valores del centro de gravedad y matrices de inercia del eslabón 1.

Licencia Creative Commons:
 Atribución - No Comercial - Compartir Igual 3.0 Venezuela
 (CC BY-NC-SA 3.0 VE)

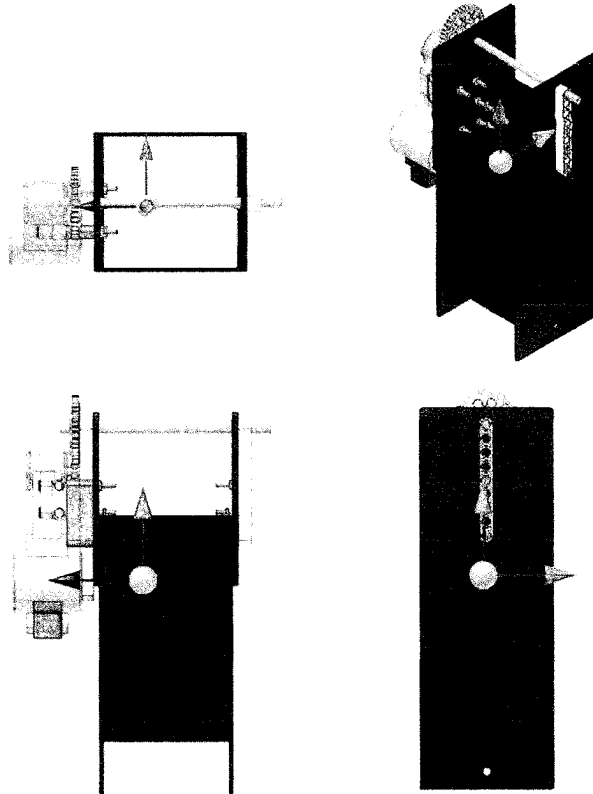


Figura 91 Centro de gravedad del eslabón 2

WWW.BDIGITAL.ULA.VE

General Properties

Include Cosmetic Welds

Include QTY Overrides

Center of Gravity*

Mass 0,473 kg (Relative)

X -86,769 mm (Relat

Area 170517,530 mm²

Y -1,732 mm (Relativ

Volume 228832,583 mm³

Z 13,092 mm (Relativ

Inertial Properties*

Principal

Global

Center Of Gravity

Mass Moments

Ixx 1059,248 kg m

Calculated using negative integral.

Ixy 1,678 kg mm²

Iyy 2089,822 kg m

Ixz -99,320 kg mm

Iyz 43,420 kg mm²

Izz 1594,025 kg m

*Values do not reflect user-overridden mass or volume

Figura 92 Valores del centro de gravedad y matrices de inercia del eslabón 2.

Licencia Creative Commons:
 Atribución - No Comercial - Compartir Igual 3.0 Venezuela
 (CC BY-NC-SA 3.0 VE)

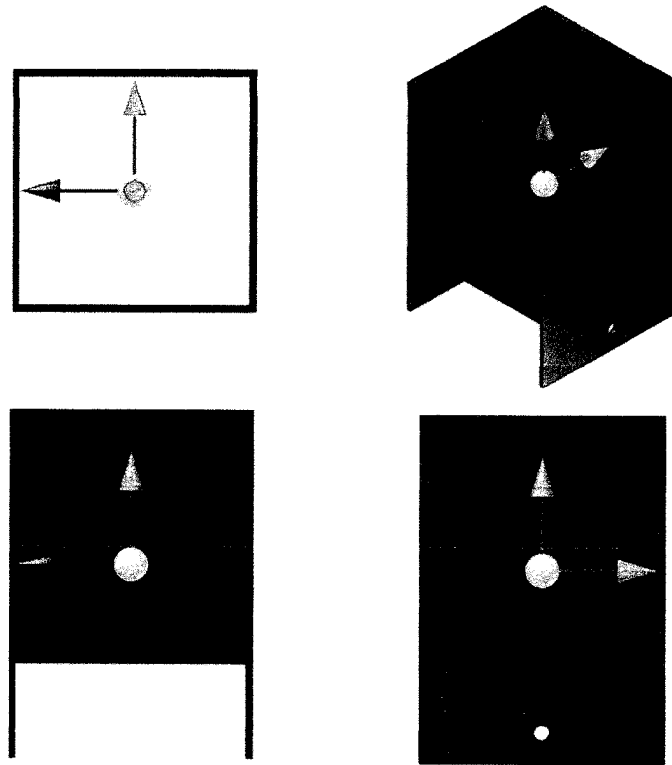


Figura 93 Centro de gravedad del eslabón 3.

WWW.BDIGITAL.ULA.VE

General Properties

Include Cosmetic Welds Include QTY Overrides

Center of Gravity

Mass	0,153 kg (Relative)	X	-78,809 mm (Relat
Area	58043,492 mm ²	Y	-0,000 mm (Relativ
Volume	56553,460 mm ³	Z	-0,000 mm (Relativ

Inertial Properties

Mass Moments

Ixx	282,163 kg mm ²	Calculated using negative integral.			
Ixy	0,000 kg mm ²	Iyy	284,792 kg mm ²		
Ixz	0,000 kg mm ²	Iyz	-0,000 kg mm ²	Izz	262,959 kg mm ²

Figura 94 Valores del centro de gravedad y matrices de inercia del eslabón 3.

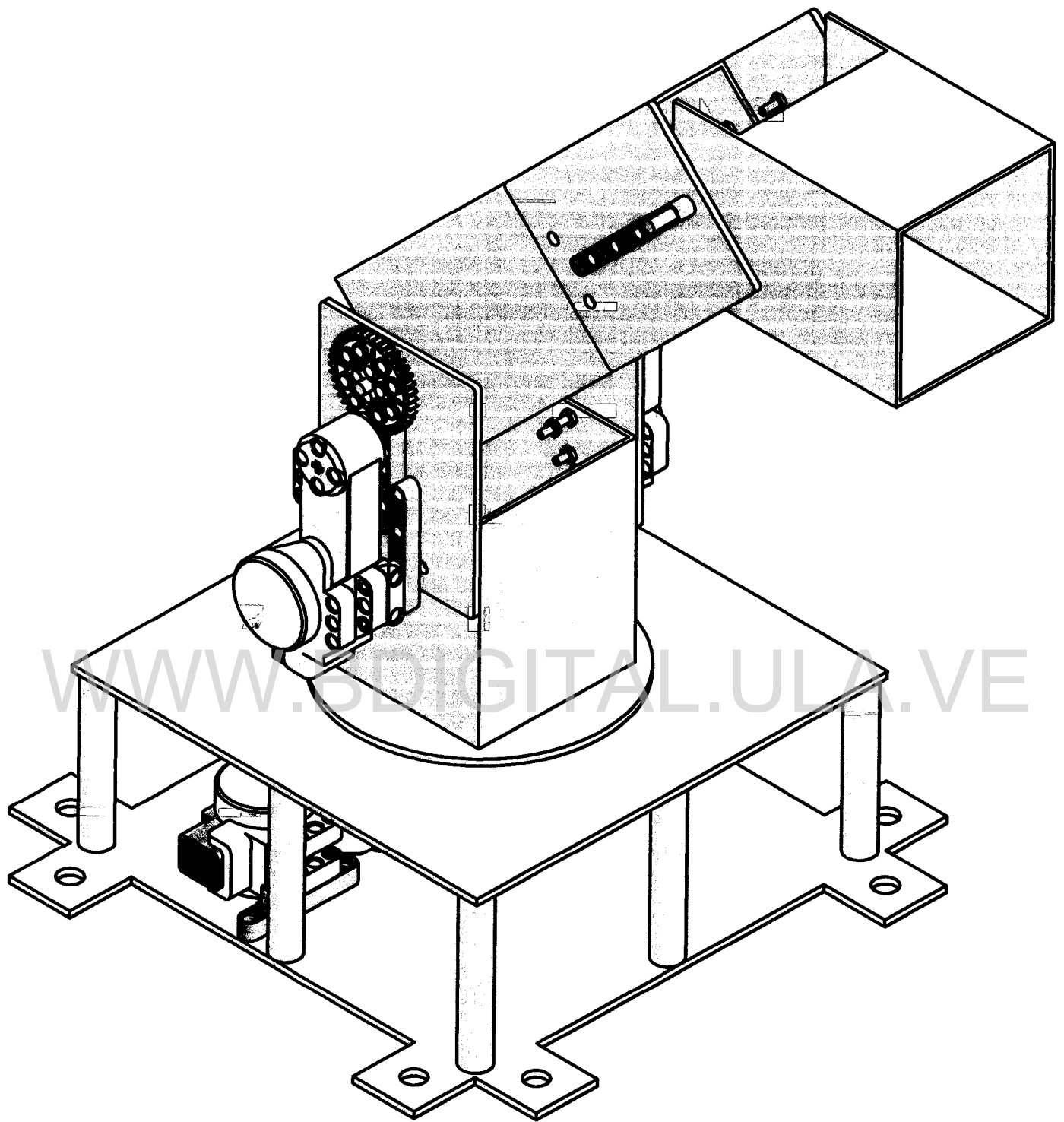
Licencia Creative Commons:
 Atribución - No Comercial - Compartir Igual 3.0 Venezuela
 (CC BY-NC-SA 3.0 VE)



Anexo N

Documentación Técnica del Robot Manipulador propuesto

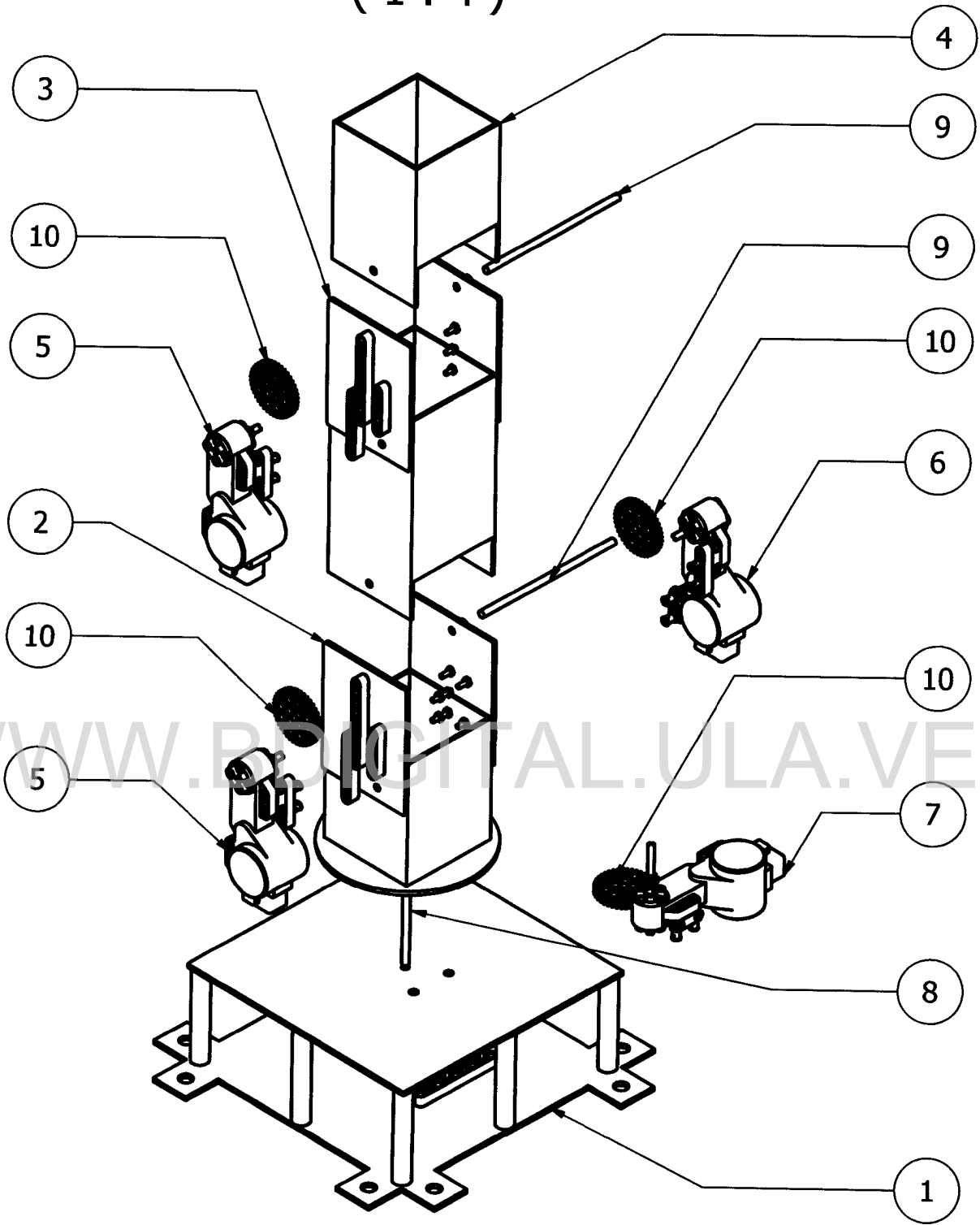
WWW.BDIGITAL.ULA.VE



Licencia Creative Commons:
Atribución - No Comercial - Compartir Igual 3.0 Venezuela
(CC BY-NC-SA 3.0 VE)



Diseñado por: Luis Andueza	Especialización y Maestría en Ingeniería de Control y Automatización		Fecha: 22/03/2008	Unidades: (mm)
 UNIVERSIDAD DE LOS ANDES VENEZUELA		Brazo Manipulador		
		Numero de Parte: Brazo manipulador	Escala Indicada	Hoja 1 / 28

(1 : 4)





Diseñado por: Luis Andueza	Especialización y Maestría en Ingeniería de Control y Automatización		Fecha: 22/03/2008	Unidades: (mm)
 UNIVERSIDAD DE LOS ANDES VENEZUELA	Brazo Manipulador			
Número de Parte: Brazo manipulador			Escala Indicada	Hoja 2 / 28

Parts List

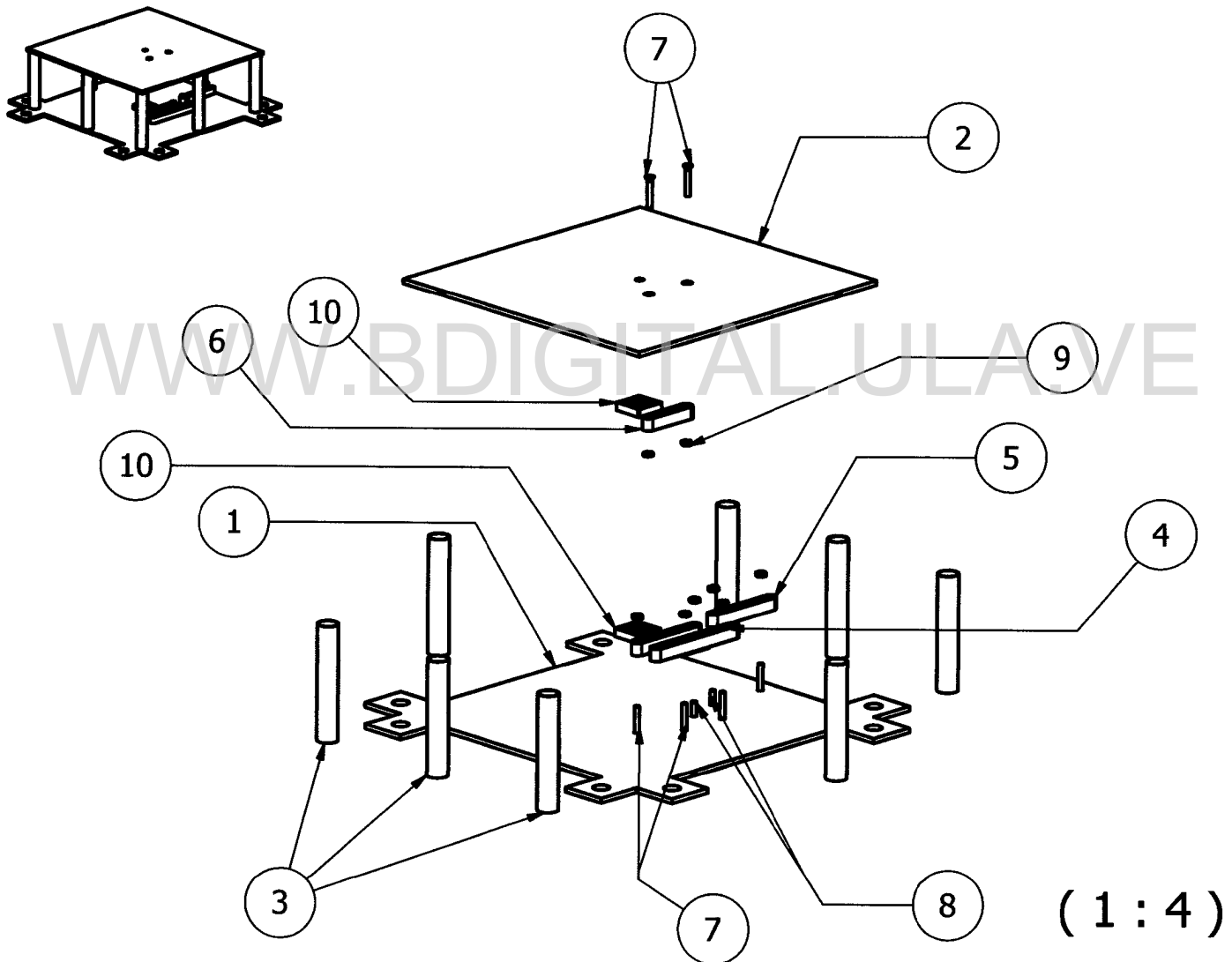
ITEM	QTY	PART NUMBER	DESCRIPTION	MATERIAL
1	1	Eslabon base	Base del brazo manipulador	
2	1	Eslabon uno	Primer eslabón	
3	1	Eslabon dos	Segundo eslabón	
4	1	Eslabon tres	Tercer eslabón	
5	2	Motor derecho	Ensamblaje motor lado derecho	
6	1	Motor izquierdo	Ensamblaje motor lado izquierdo	
7	1	Motor base	Ensamblaje motor para la base	
8	1	Eje 02	Eje de la base	Aluminio
9	2	Eje 01	Eje entre eslabones	Aluminio
10	4	Engranaje 40	Gear Wheel T40	Plástico



WWW.BDIGITAL.ULA.VE

Diseñado por: Luis Andueza	Especialización y Maestría en Ingeniería de Control y Automatización		Fecha: 22/03/2008	Unidades: (mm)
 UNIVERSIDAD DE LOS ANDES VENEZUELA		Brazo Manipulador		
Licencia Creative Commons: Atribución - No Comercial - Compartir Igual 3.0 Venezuela (CC BY-NC-SA 3.0 VE)			Escala Indicada 3 / 28	Hoja 3 / 28

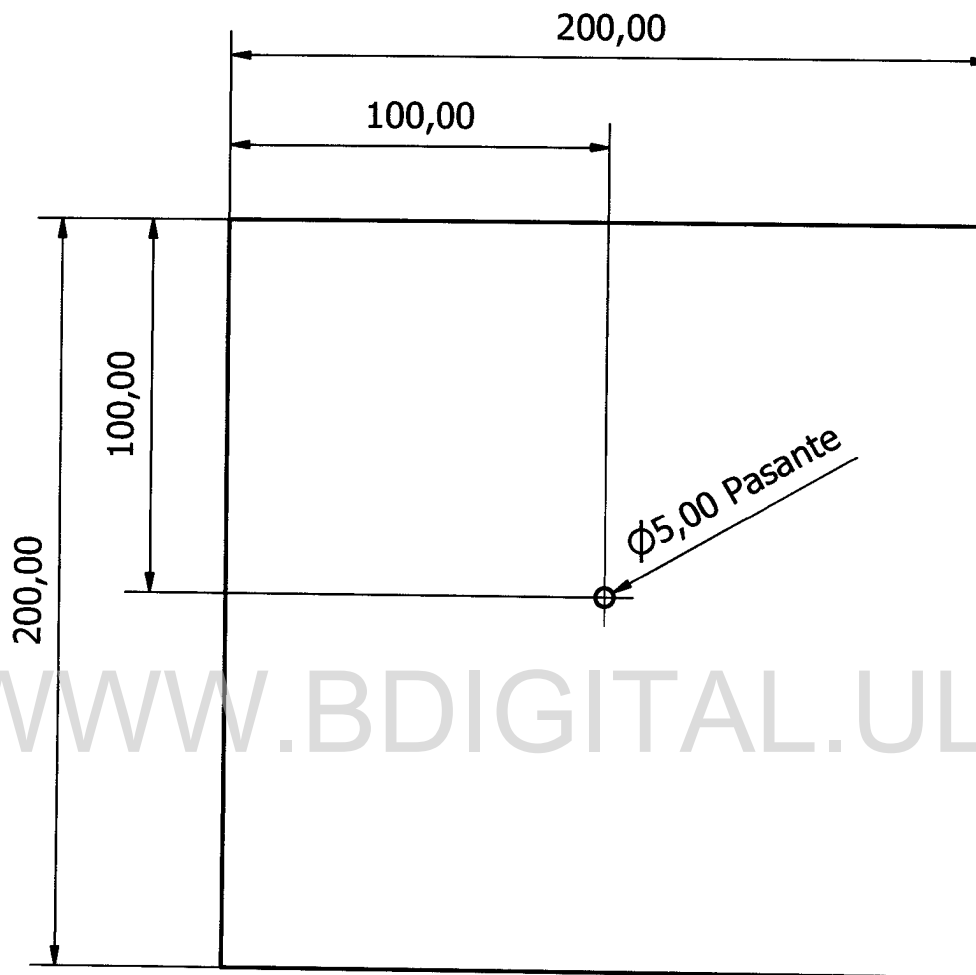
Parts List

ITEM	QTY	PART NUMBER	DESCRIPTION	MATERIAL
1	1	Base A	Tapa inferior	Aluminio
2	1	Base B	Tapa superior	Aluminio
3	8	Tubo 1/2"	Perfil redondo 1/2"	Aluminio
4	1	Columna 9	Technic Liftarm 9 LEGO	Plástico
5	2	Columna 7	Technic Liftarm 7 LEGO	Plástico
6	1	Columna 5	Technic Liftarm 5 LEGO	Plástico
7	6	AS 1427 - M3 x 20	ISO metric machine screws	Acero
8	2	AS 1427 - M3 x 12	ISO metric machine screws	Acero
9	8	AS 1474 - M3	Hex Nut	Acero
10	2	Chumacera	Ensamblaje rodamiento y carcasa	





Diseñado por: Luis Andueza	Especialización y Maestría en Ingeniería de Control y Automatización	 Fecha: 22/03/2008	Unidades: (mm)
 UNIVERSIDAD DE LOS ANDES VENEZUELA		Brazo Manipulador Número de Hoja: Eslabon base	
		Escala Indicada	Hoja 4 / 28

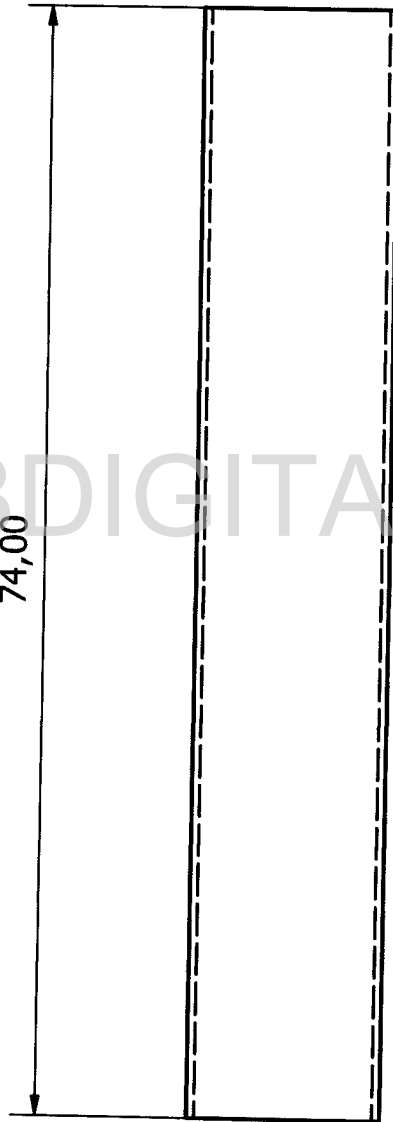
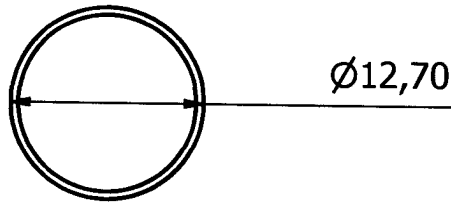
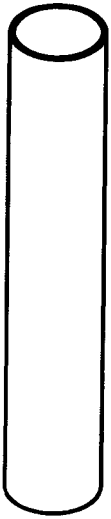
(1 : 2)



Espesor 3 mm

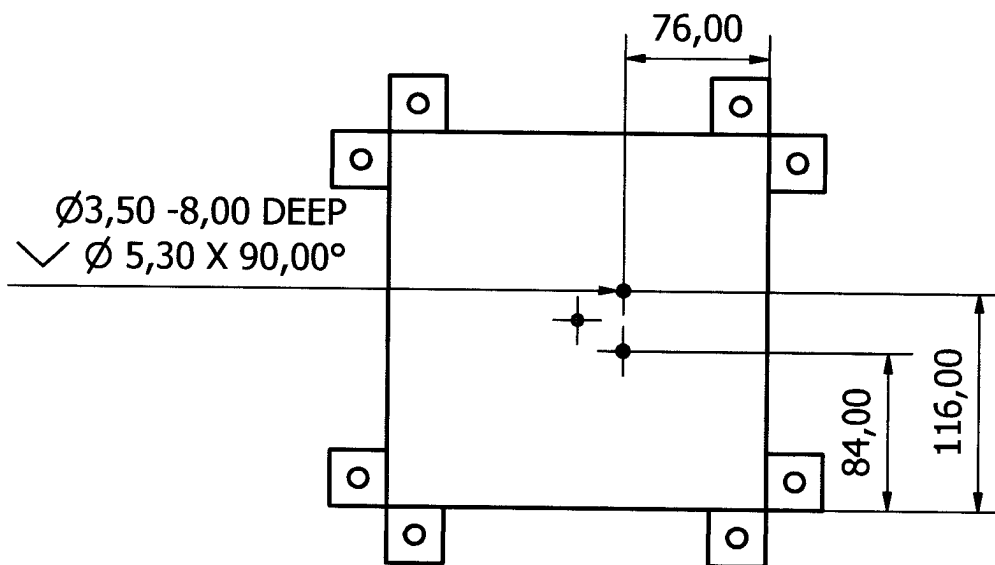
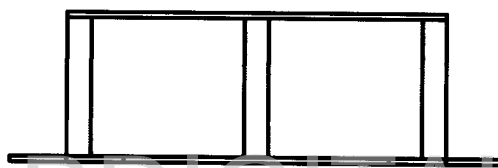
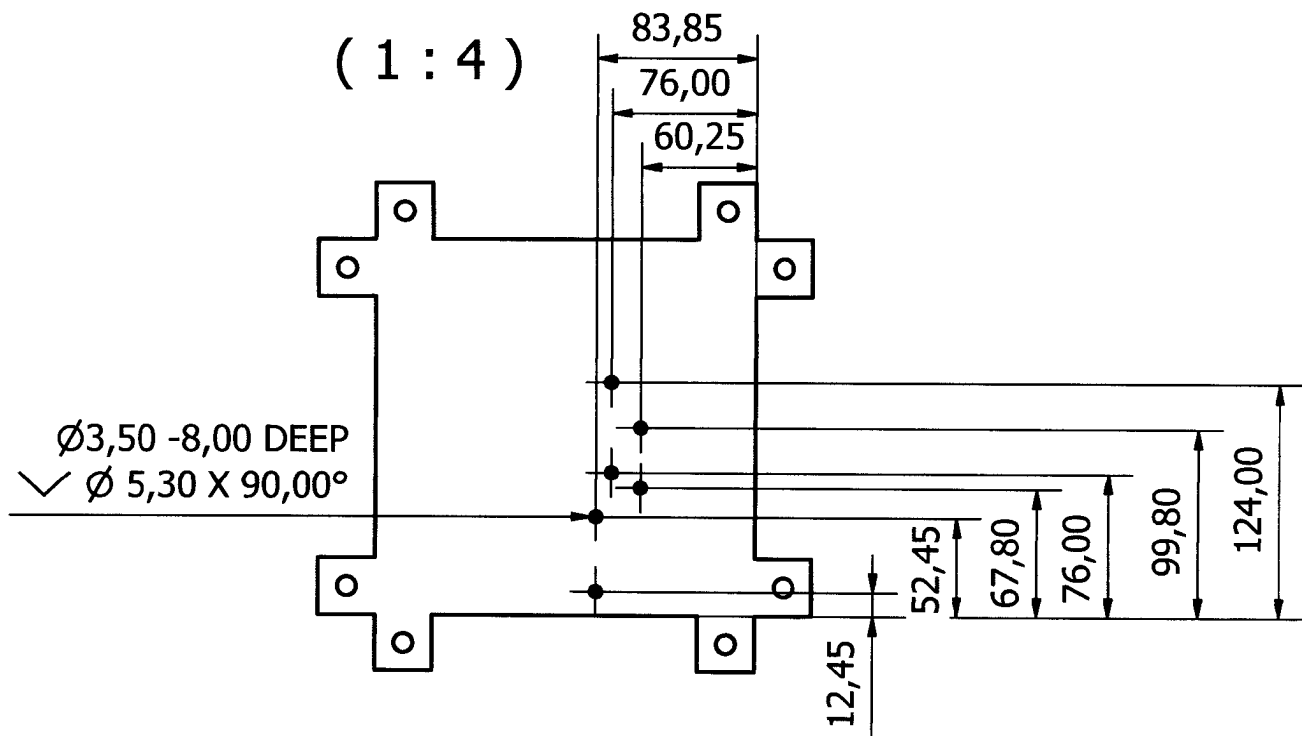
Diseñado por: Luis Andueza	Especialización y Maestría en Ingeniería de Control y Automatización		Fecha: 22/03/2008	Unidades: (mm)
 UNIVERSIDAD DE LOS ANDES VENEZUELA	Brazo Manipulador		Escala Indicada	Hoja 6 / 28



(2 : 1)



WWW.BDIGITAL.ULA.VE

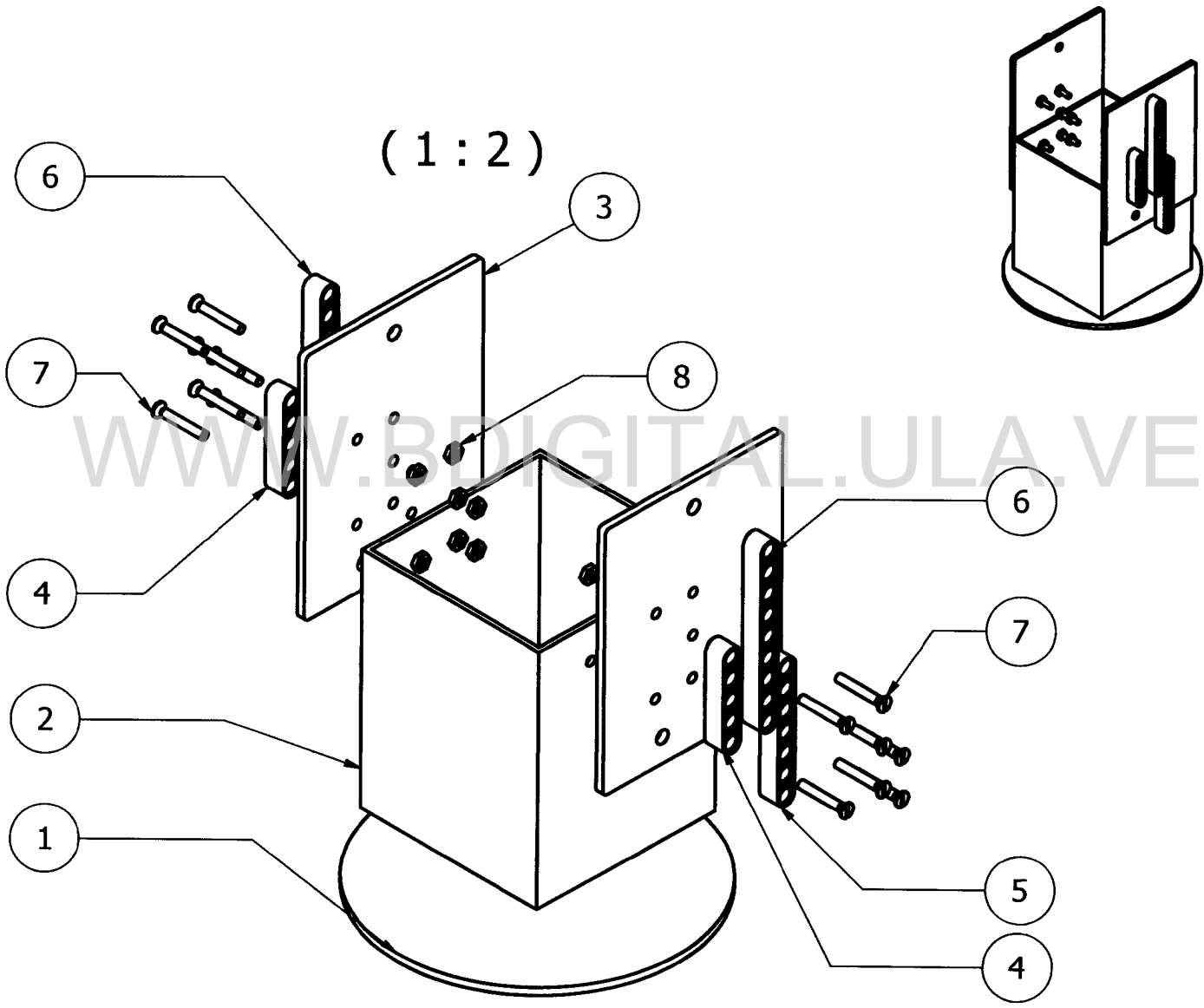
Diseñado por: Luis Andueza	Especialización y Maestría en Ingeniería de Control y Automatización		Fecha: 22/03/2008	Unidades: (mm)
	UNIVERSIDAD DE LOS ANDES VENEZUELA	Brazo Manipulador		
Número de Parte Tubo 1/2"			Escala Indicada	Hoja 7 / 28



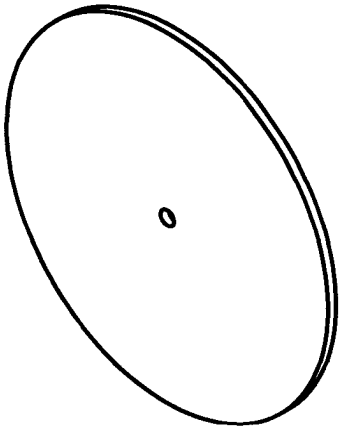
Diseñado por: Luis Andueza	Especialización y Maestría en Ingeniería de Control y Automatización		Fecha: 22/03/2008	Unidades: (mm)
 UNIVERSIDAD DE LOS ANDES VENEZUELA		Brazo Manipulador		
Licencia Creative Commons: Eslabon base		Escala Indicada	Hoja 8 / 28	

Parts List

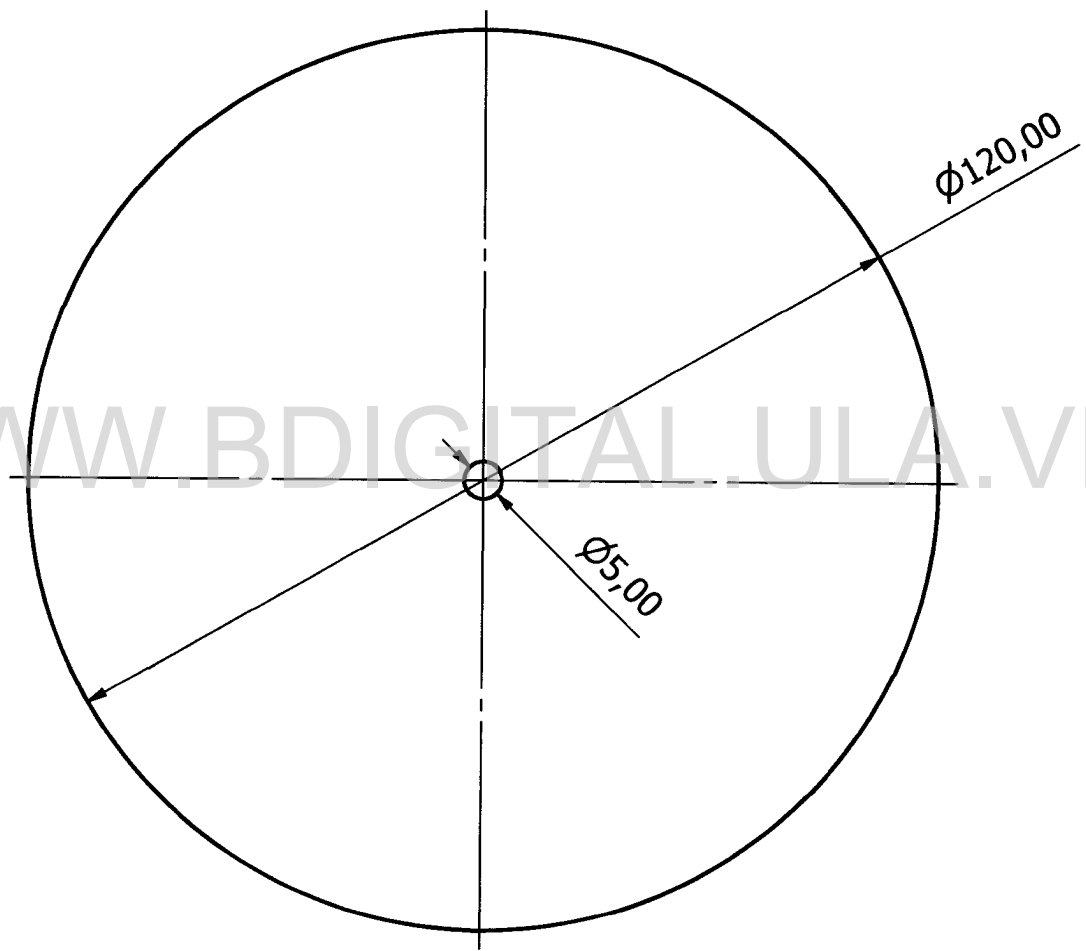
ITEM	QTY	PART NUMBER	DESCRIPTION	MATERIAL
1	1	Base E01	Base del eslabón uno	Aluminio
2	1	Eslabon Uno	Perfil cuadrado 3"	Aluminio
3	2	Extensión	Extensiones de los eslabones	Aluminio
4	2	Columna 5	Technic Liftarm 5 LEGO	Plástico
5	2	Columna 7	Technic Liftarm 7 LEGO	Plástico
6	2	Columna 9	Technic Liftarm 9 LEGO	Plástico
7	14	AS 1427 - M3 x 20	ISO metric machine screws	Acero
8	14	AS 1474 - M3	Hex Nut	Acero





Diseñado por: Luis Andueza	Especialización y Maestría en Ingeniería de Control y Automatización		Fecha: 22/03/2008	Unidades: (mm)
UNIVERSIDAD DE LOS ANDES VENEZUELA		Brazo Manipulador		
Numero de Parte Eslabon uno		Escala Indicada	Hoja 9 / 28	

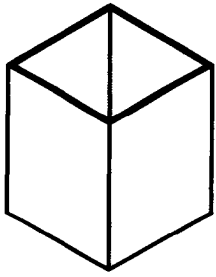


(1 : 1)

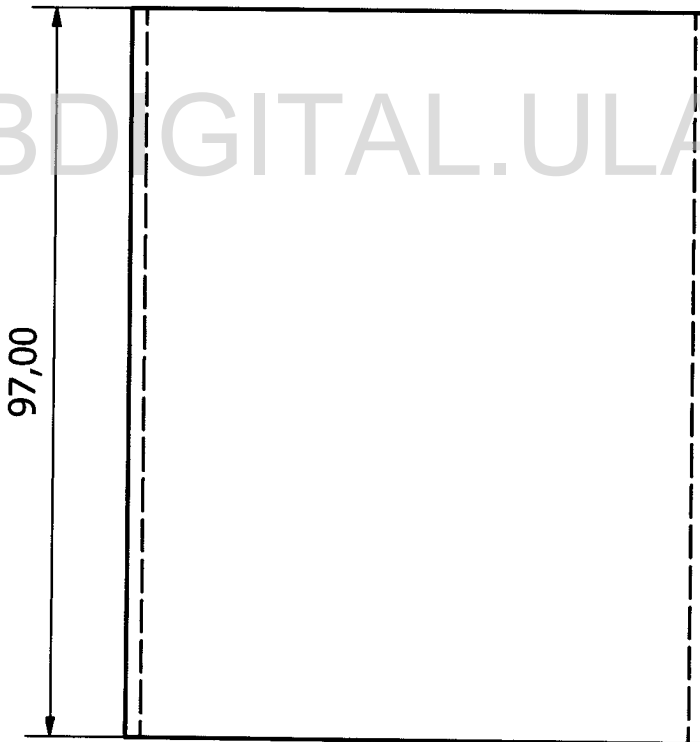
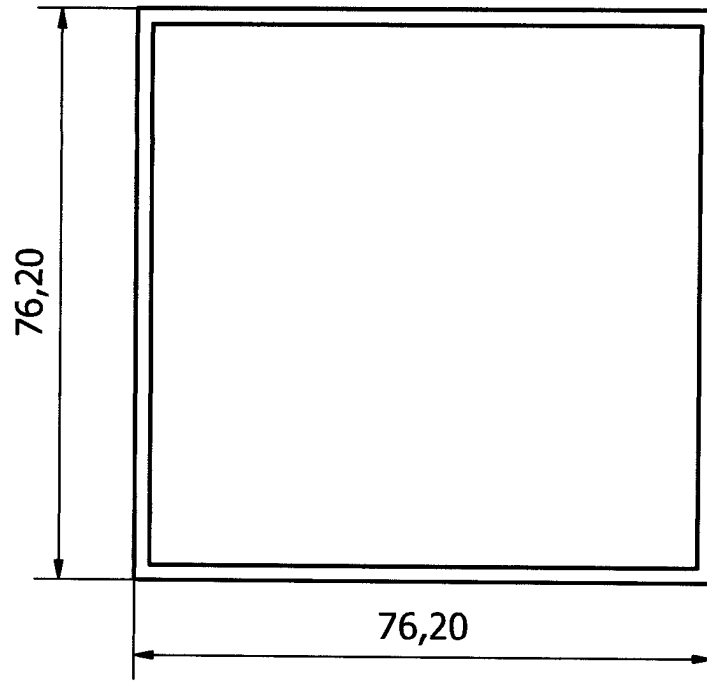


Espesor 3 mm



Diseñado por: Luis Andueza	Especialización y Maestría en Ingeniería de Control y Automatización		Fecha: 22/03/2008	Unidades: (mm)
 UNIVERSIDAD DE LOS ANDES VENEZUELA		Brazo Manipulador		
Numero de Parte Base E01			Escala Indicada	Hoja 10 / 28

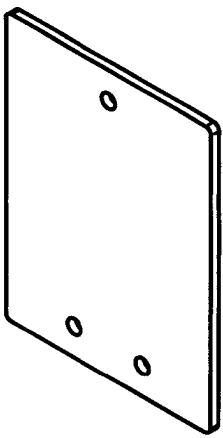


(1 : 1)

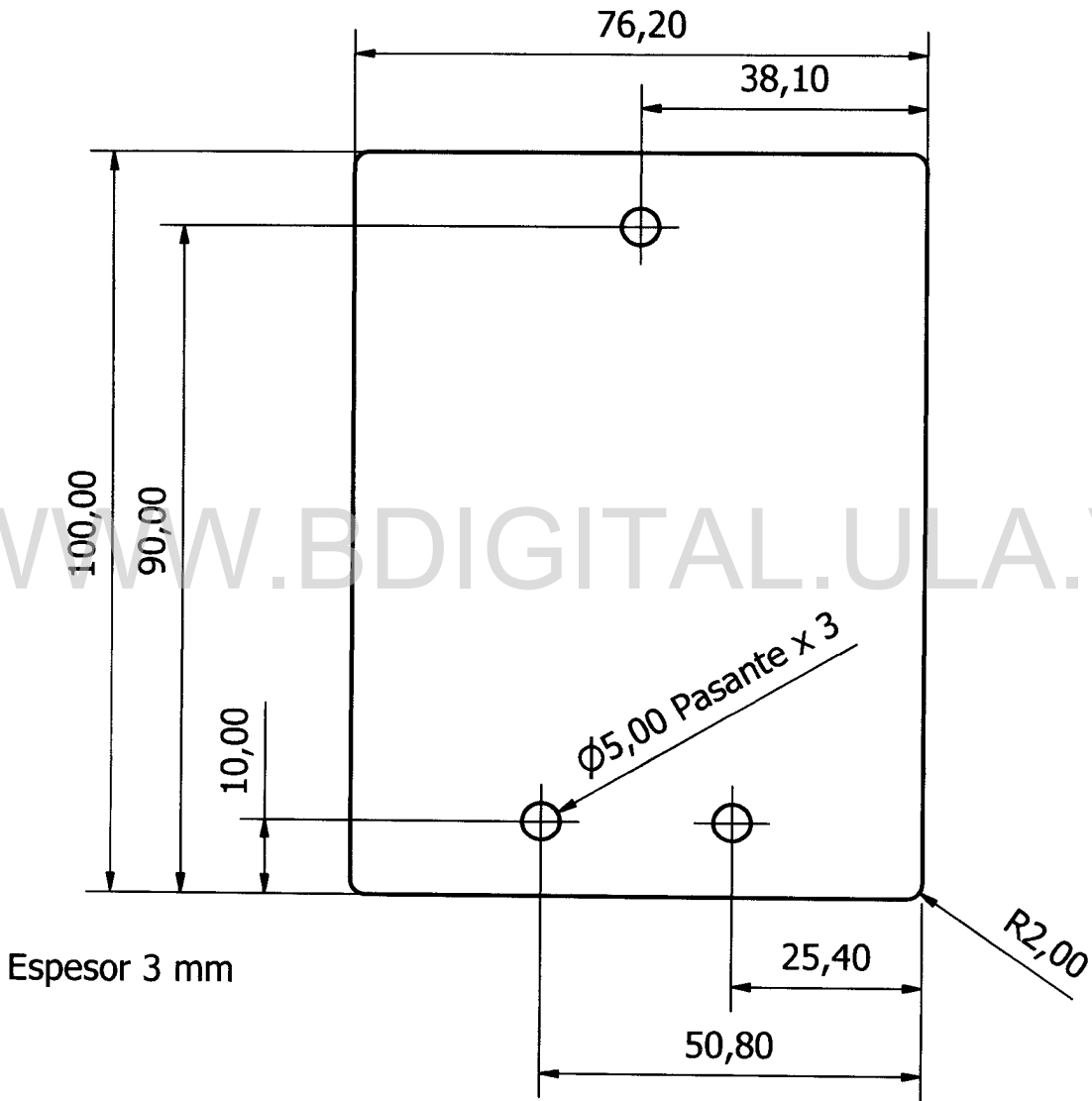


WWW.BDIGITAL.ULA.VE

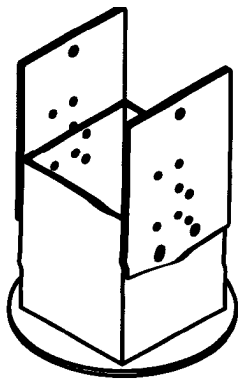
Diseñado por: Luis Andueza	Especialización y Maestría en Ingeniería de Control y Automatización		Fecha: 22/03/2008	Unidades: (mm)
	UNIVERSIDAD DE LOS ANDES VENEZUELA	Brazo Manipulador		
Número de Parte: Eslabon Uno		Escala Indicada	Hoja 11 / 28	



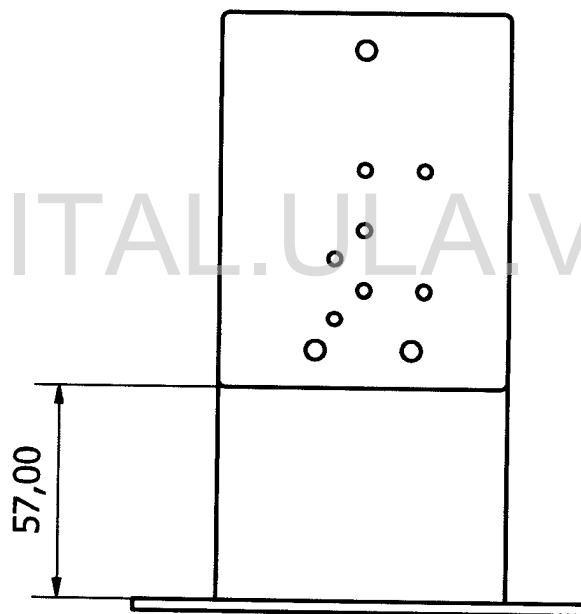
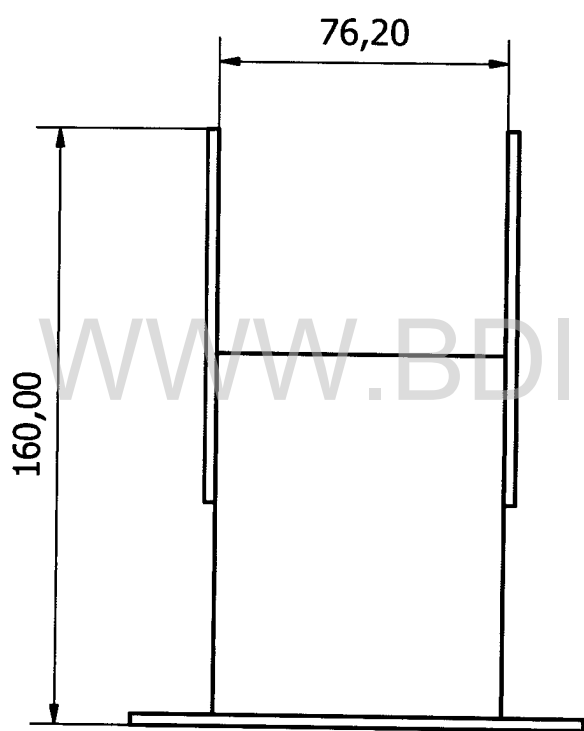
(1 : 1)



Diseñado por: Luis Andueza	Especialización y Maestría en Ingeniería de Control y Automatización		Fecha: 22/03/2008	Unidades: (mm)
	UNIVERSIDAD DE LOS ANDES VENEZUELA	Brazo Manipulador		
Número de Parte: Extensión		Escala Indicada		Hoja 12 / 28

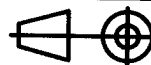


(1 : 2)



Diseñado por:
Luis Andueza

Especialización y Maestría en Ingeniería
de Control y Automatización



Fecha:
22/03/2008

Unidades:
(mm)



UNIVERSIDAD
DE LOS ANDES
VENEZUELA

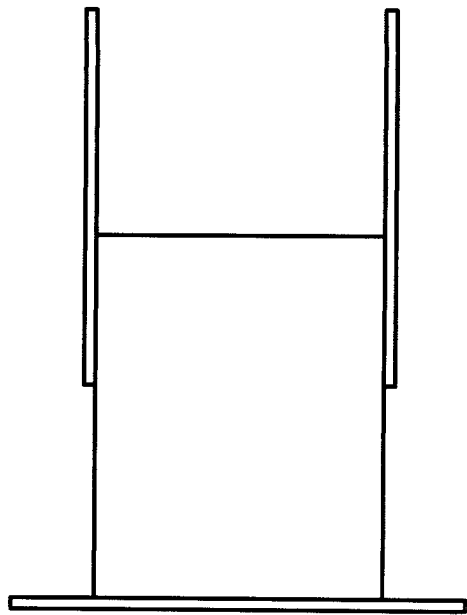
Brazo Manipulador

Numero de Parte
Eslabon uno

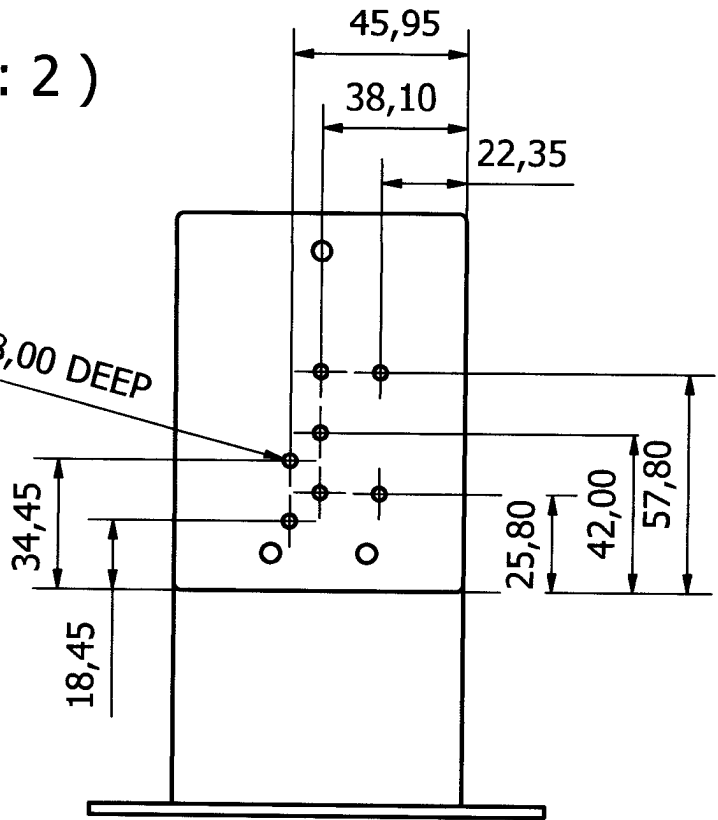
Escala
Indicada

Hoja
13 / 28

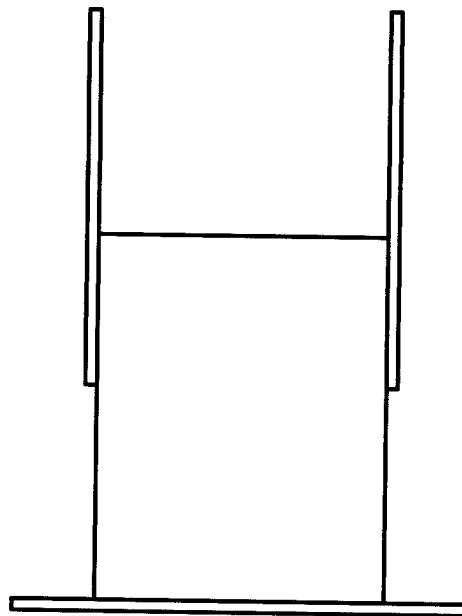
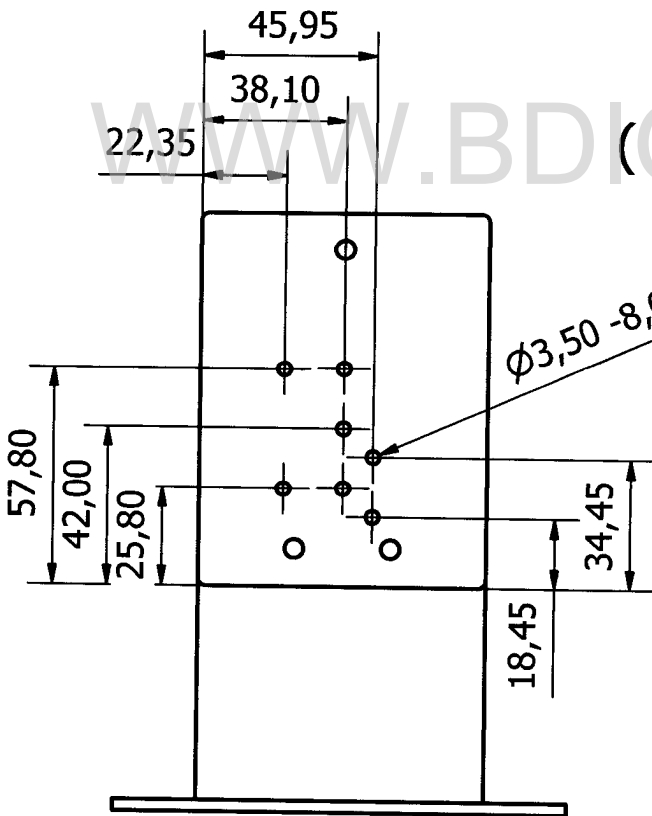
(1 : 2)



Ø3,50 -8,00 DEEP



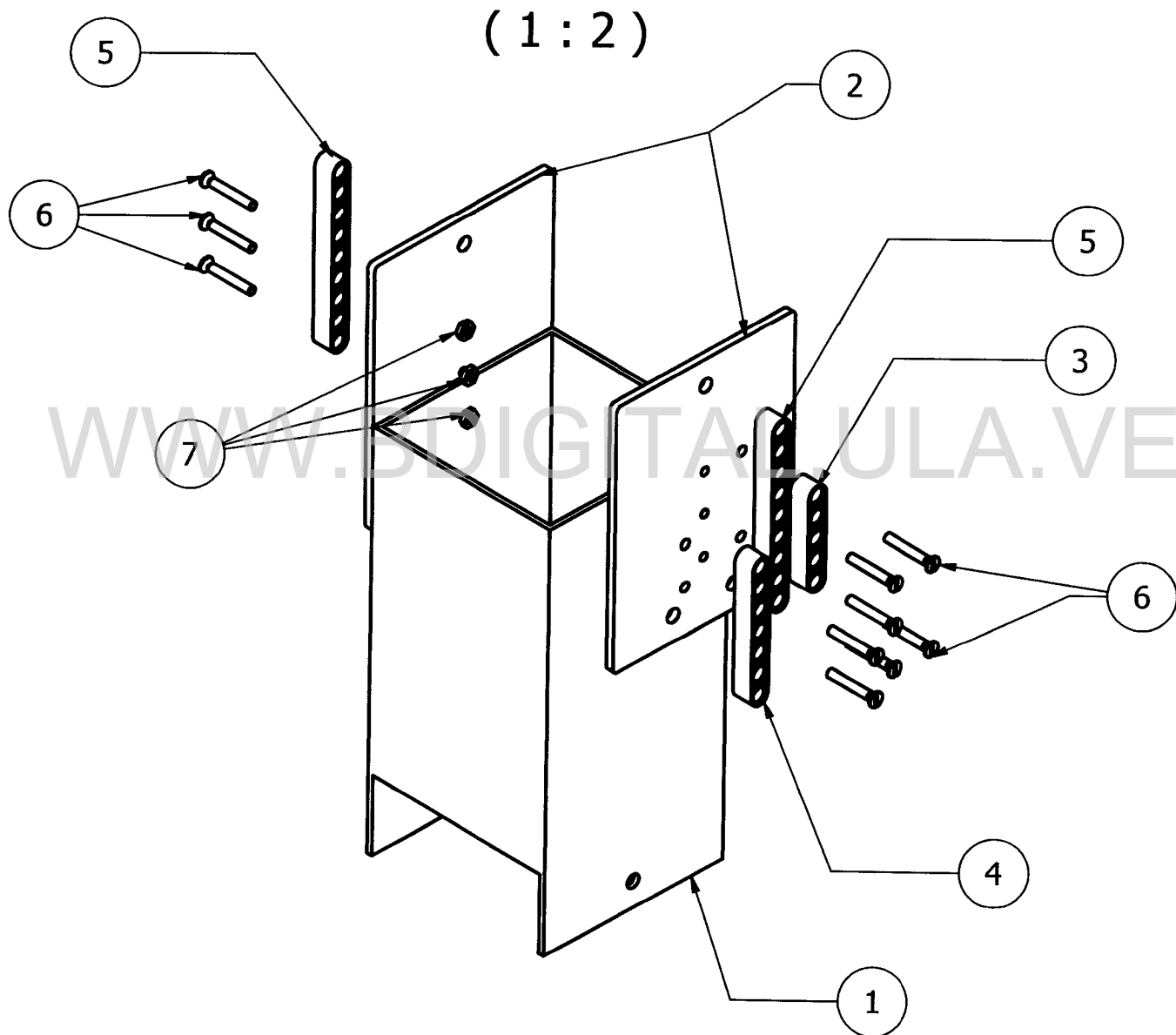
(1 : 2)

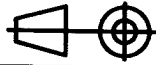



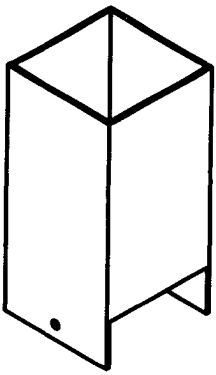
Diseñado por: Luis Andueza	Especialización y Maestría en Ingeniería de Control y Automatización		Fecha: 22/03/2008	Unidades: (mm)
		Brazo Manipulador		
UNIVERSIDAD DE LOS ANDES VENEZUELA		Número de Parte: Eslabon uno	Escala Indicada	Hoja 14 / 28

Parts List

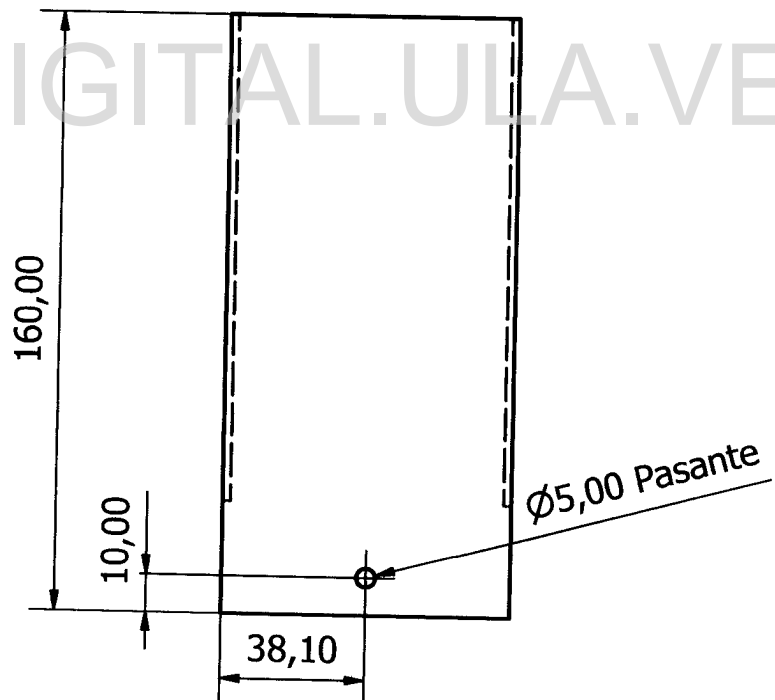
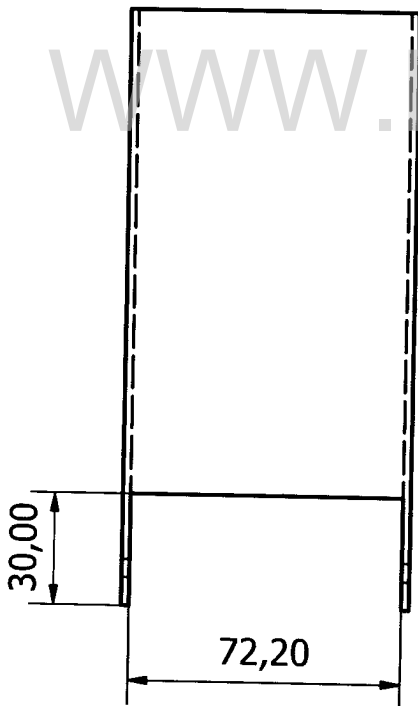
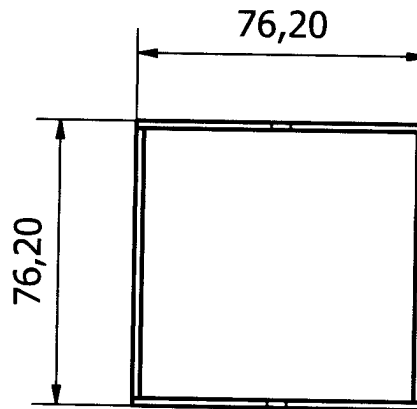
ITEM	QTY	PART NUMBER	DESCRIPTION	MATERIAL
1	1	Eslabon dos	Perfil cuadrado 3"	Aluminio
2	2	Extensión	Extensiones de los eslabones	Aluminio
3	1	Columna 5	Technic Liftarm 5 LEGO	Plástico
4	1	Columna 7	Technic Liftarm 7 LEGO	Plástico
5	2	Columna 9	Technic Liftarm 9 LEGO	Plástico
6	10	AS 1427 - M3 x 20	ISO metric machine screws	Acero
7	10	AS 1474 - M3	Hex Nut	Acero





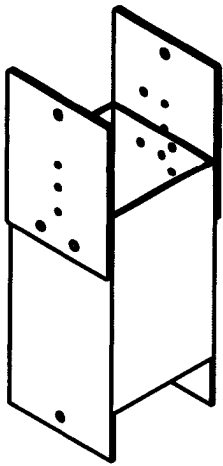
Diseñado por: Luis Andueza	Especialización y Maestría en Ingeniería de Control y Automatización		Fecha: 22/03/2008	Unidades: (mm)
 UNIVERSIDAD DE LOS ANDES VENEZUELA		Brazo Manipulador		
Número de Parte: Eslabon dos		Escala Indicada		Hoja 15 / 28



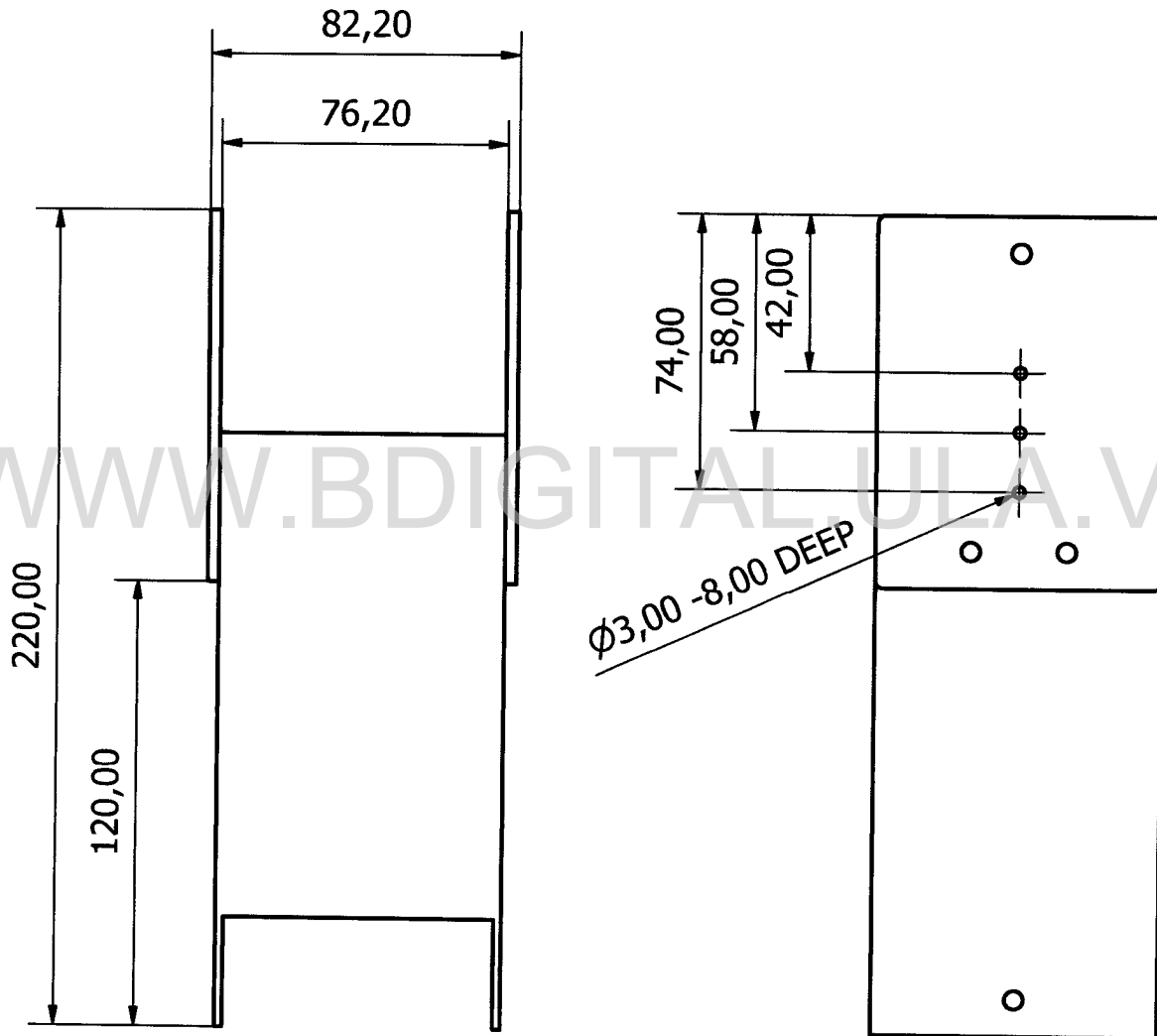
(1 : 2)

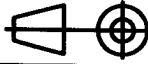



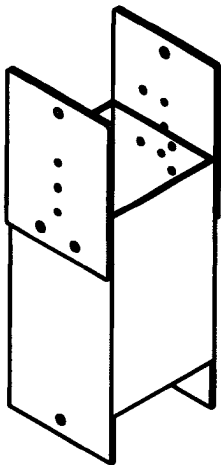
Diseñado por: Luis Andueza	Especialización y Maestría en Ingeniería de Control y Automatización		Fecha: 22/03/2008	Unidades: (mm)
 UNIVERSIDAD DE LOS ANDES VENEZUELA		Brazo Manipulador		
Número de Parte: Esiabon dos		Escala Indicada	Hoja 16 / 28	



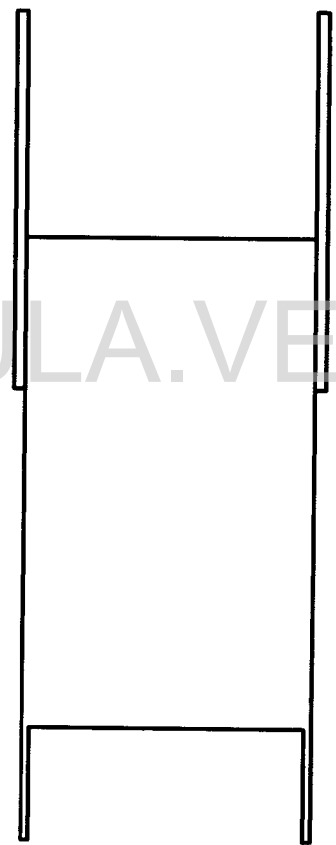
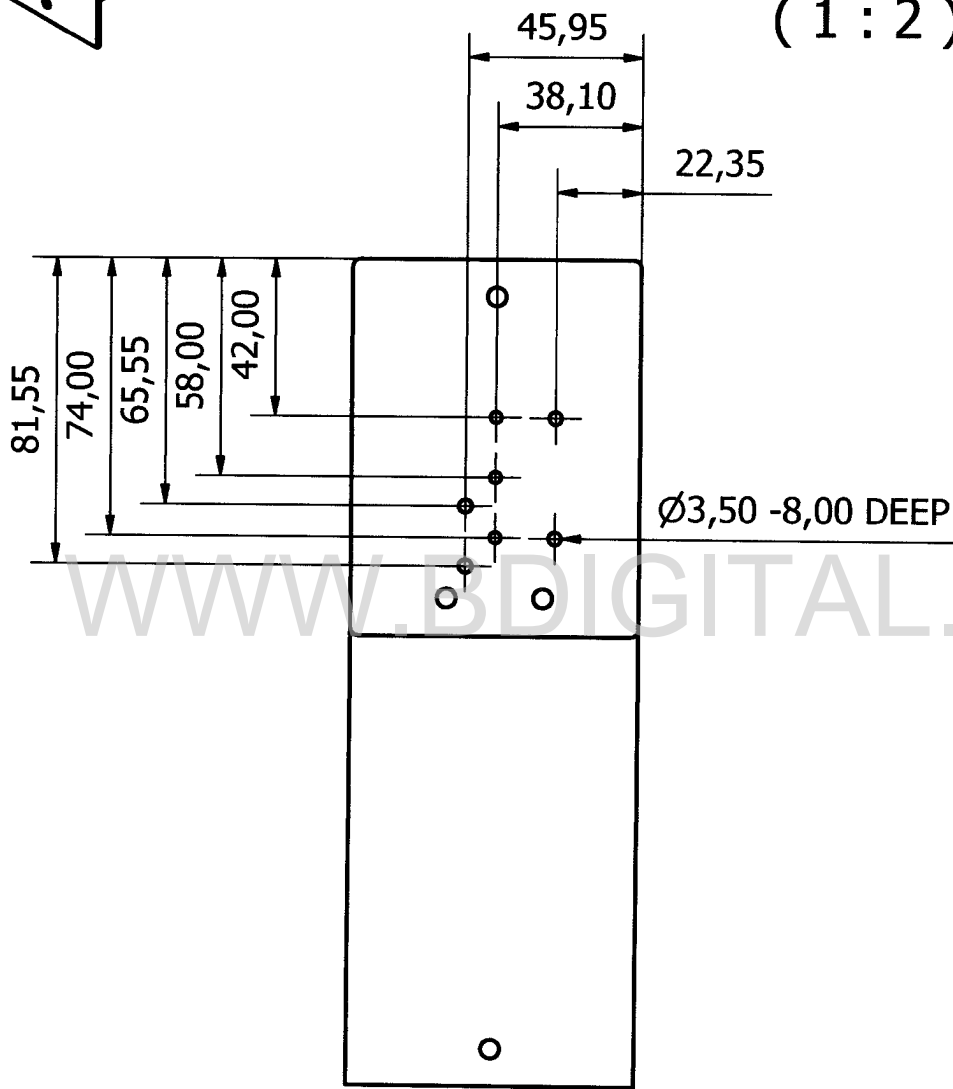
(1 : 2)



Diseñado por: Luis Andueza	Especialización y Maestría en Ingeniería de Control y Automatización	 Fecha: 22/03/2008	Unidades: (mm)
 UNIVERSIDAD DE LOS ANDES VENEZUELA		Brazo Manipulador	
Número de Parte: Eslabon dos		Escala Indicada	Hoja 17 / 28



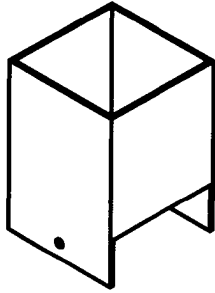
(1 : 2)



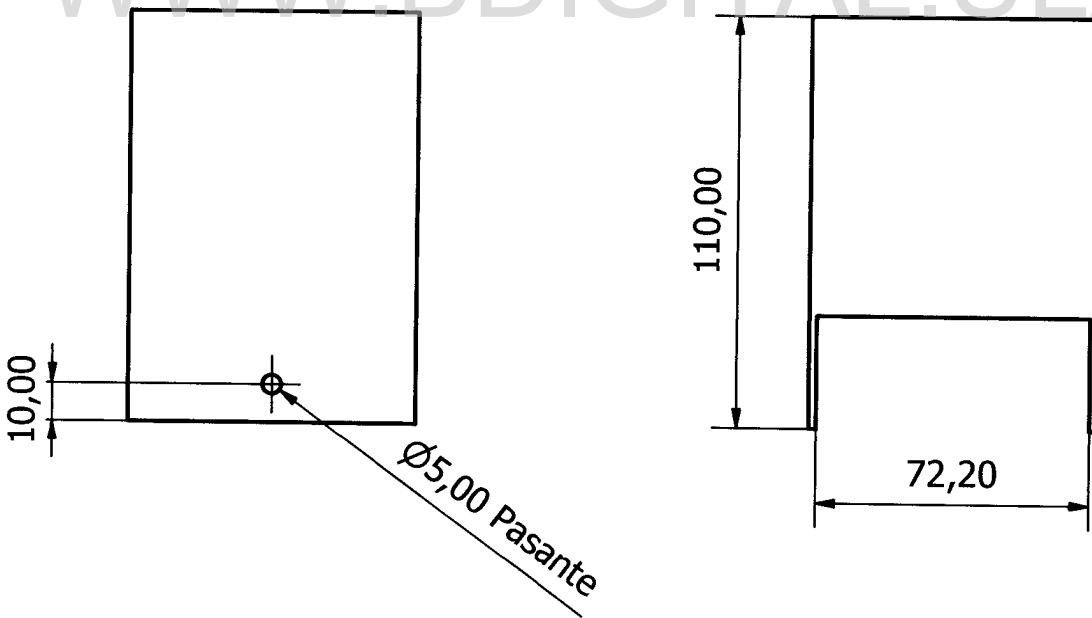
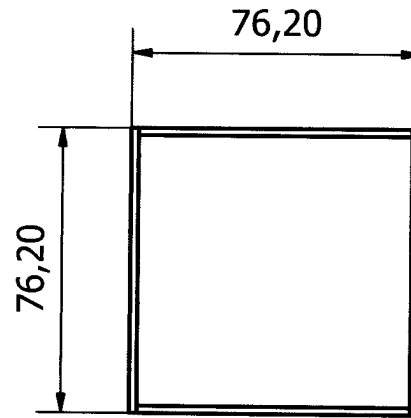
Diseñado por: Luis Andueza	Especialización y Maestría en Ingeniería de Control y Automatización		Fecha: 22/03/2008	Unidades: (mm)
UNIVERSIDAD DE LOS ANDES VENEZUELA		Brazo Manipulador		
Número de Parte: Eslabon dos		Escala Indicada	Hoja 18 / 28	

Parts List



ITEM	QTY	PART NUMBER	DESCRIPTION	MATERIAL
1	1	Eslabon 3	Perfil cuadrado 3"	Aluminio

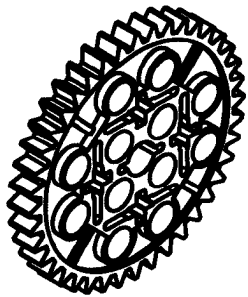


(1 : 2)

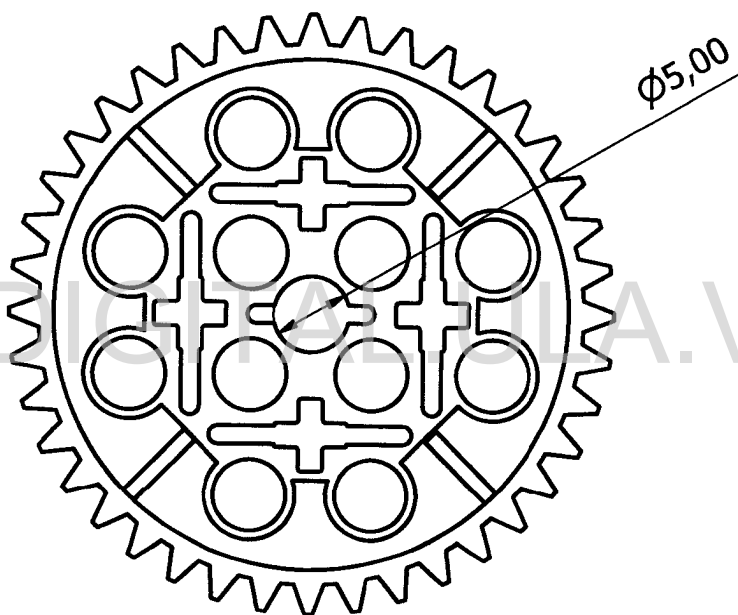


WWW.BDIGITAL.ULA.VE



Diseñado por: Luis Andueza	Especialización y Maestría en Ingeniería de Control y Automatización		Fecha: 22/03/2008	Unidades: (mm)
 UNIVERSIDAD DE LOS ANDES VENEZUELA		Brazo Manipulador		
Número de Parte: Eslabon tres		Escala Indicada	Hoja 19 / 28	



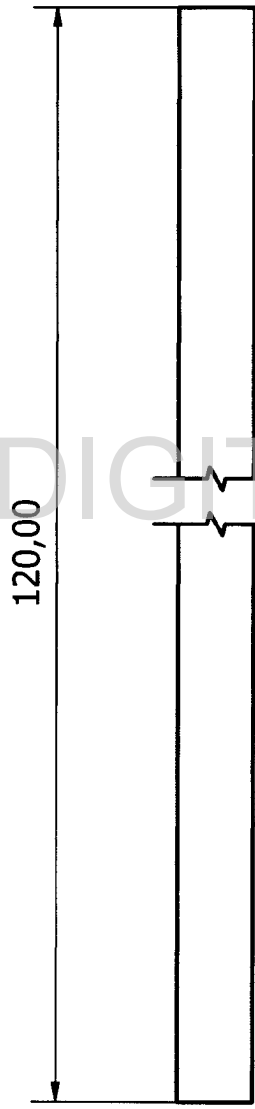
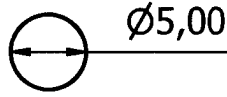
(2 : 1)





WWW.BENEFICIALIA.VE

Diseñado por: Luis Andueza	Especialización y Maestría en Ingeniería de Control y Automatización		Fecha: 22/03/2008	Unidades: (mm)
	UNIVERSIDAD DE LOS ANDES VENEZUELA	Brazo Manipulador		
		Numero de Parte Engranaje 40	Escala Indicada	Hoja 20 / 28

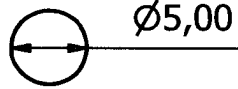
(2 : 1)





WWW.BDIGITAL.ULA.VE

Diseñado por: Luis Andueza	Especialización y Maestría en Ingeniería de Control y Automatización		Fecha: 22/03/2008	Unidades: (mm)
	UNIVERSIDAD DE LOS ANDES VENEZUELA	Brazo Manipulador		
Número de Parte Eje 01		Escala Indicada		Hoja 21 / 28

(2 : 1)

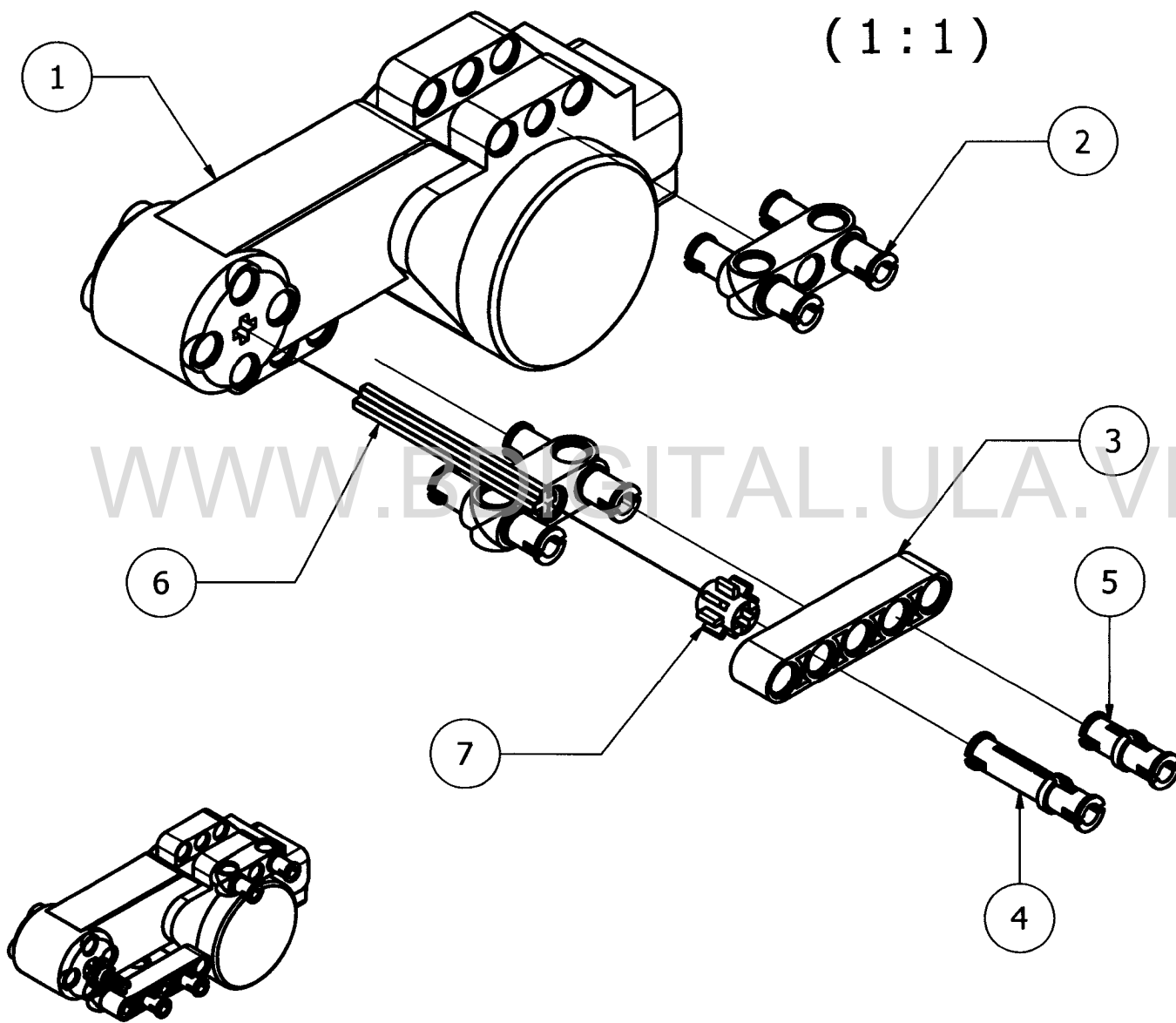




WWW.BDIGITAL.ULA.VE

Diseñado por: Luis Andueza	Especialización y Maestría en Ingeniería de Control y Automatización		Fecha: 22/03/2008	Unidades: (mm)
 UNIVERSIDAD DE LOS ANDES VENEZUELA	Brazo Manipulador			
Numero de Parte Eje 02		Escala Indicada		Hoja 22 / 28

Parts List

ITEM	QTY	PART NUMBER	DESCRIPTION
1	1	Motor LEGO NXT	Tacho Motor
2	2	Pin x 4	Beam 3 M. w. 4 Snaps
3	1	Columna 5	Technic Liftarm 5 LEGO
4	1	Pin 24	Pin 24 mm
5	1	Pin 16	Pin 16 mm
6	1	Eje 5	
7	1	Engranaje 8	Gear wheel T8 M1

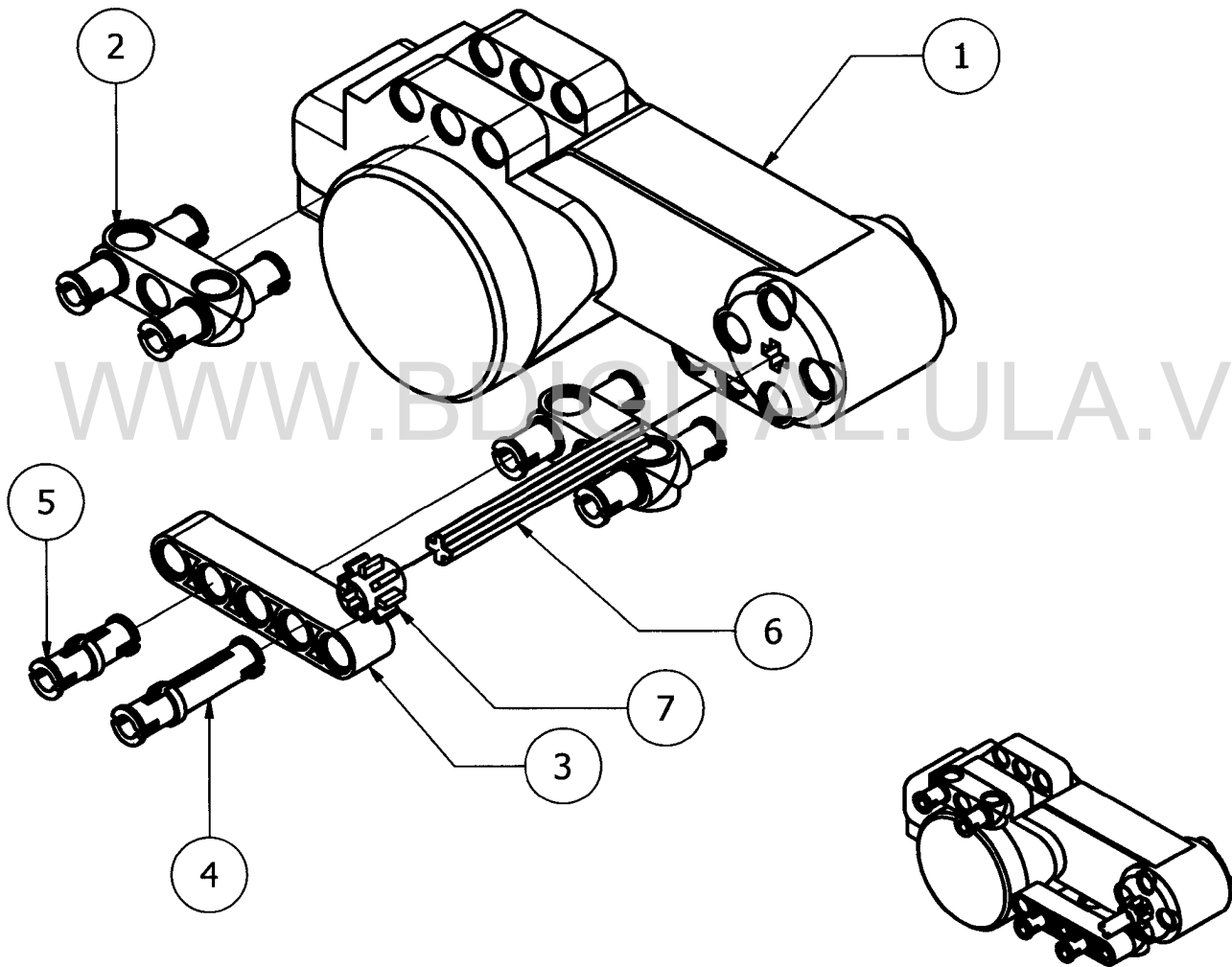


Diseñado por: Luis Andueza	Especialización y Maestría en Ingeniería de Control y Automatización		Fecha: 22/03/2008	Unidades: (mm)
 UNIVERSIDAD DE LOS ANDES VENEZUELA		Brazo Manipulador Número de Parte: Motor derecho		
Escala Indicada			Hoja 23 / 28	

Parts List

ITEM	QTY	PART NUMBER	DESCRIPTION
1	1	Motor LEGO NXT	Tacho Motor
2	2	Pin x 4	Beam 3 M. w. 4 Snaps
3	1	Columna 5	Technic Liftarm 5 LEGO
4	1	Pin 24	Pin 24 mm
5	1	Pin 16	Pin 16 mm
6	1	Eje 5	
7	1	Engranaje 8	Gear wheel T8 M1

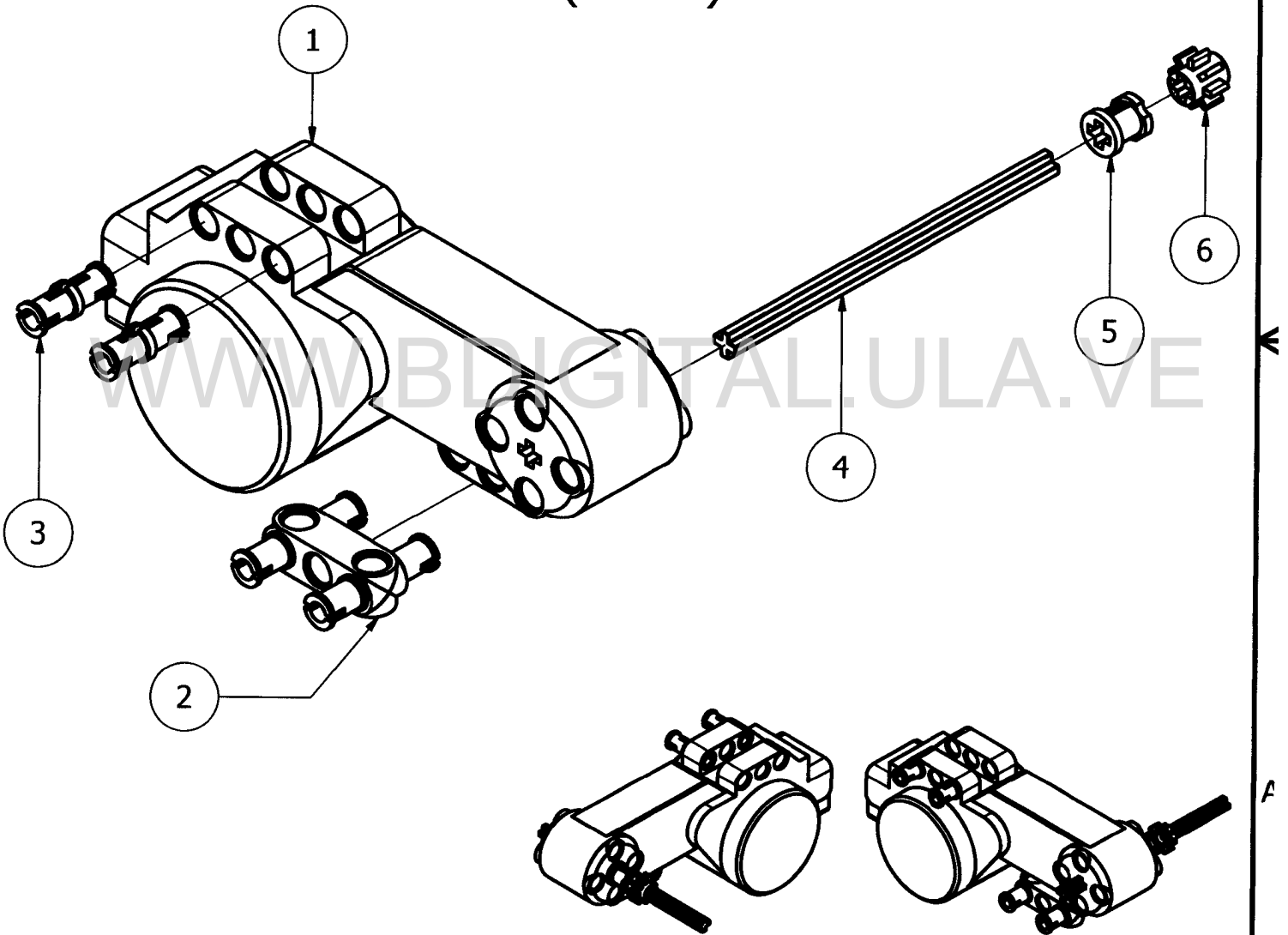
(1 : 1)





Diseñado por: Luis Andueza	Especialización y Maestría en Ingeniería de Control y Automatización		Fecha: 22/03/2008	Unidades: (mm)
 UNIVERSIDAD DE LOS ANDES VENEZUELA		Brazo Manipulador Motor izquierdo		
Escala Indicada			Hoja 24 / 28	

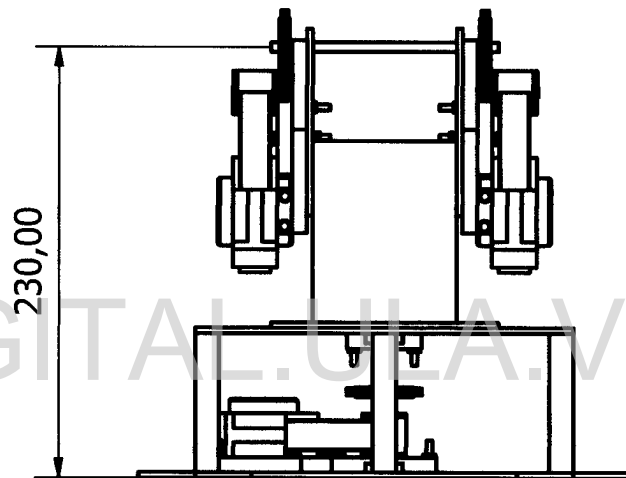
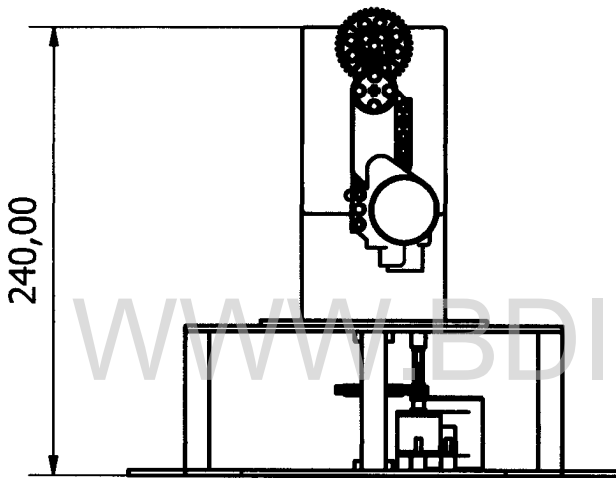
Parts List			
ITEM	QTY	PART NUMBER	DESCRIPTION
1	1	Motor LEGO NXT	Tacho Motor
2	1	Pin x 4	Beam 3 M. w. 4 Snaps
3	2	Pin 16	Pin 16 mm
4	1	Eje 9	
5	1	Separador grande	Bush for cross axle
6	1	Engranaje 8	Gear wheel T8 M1

(1 : 1)



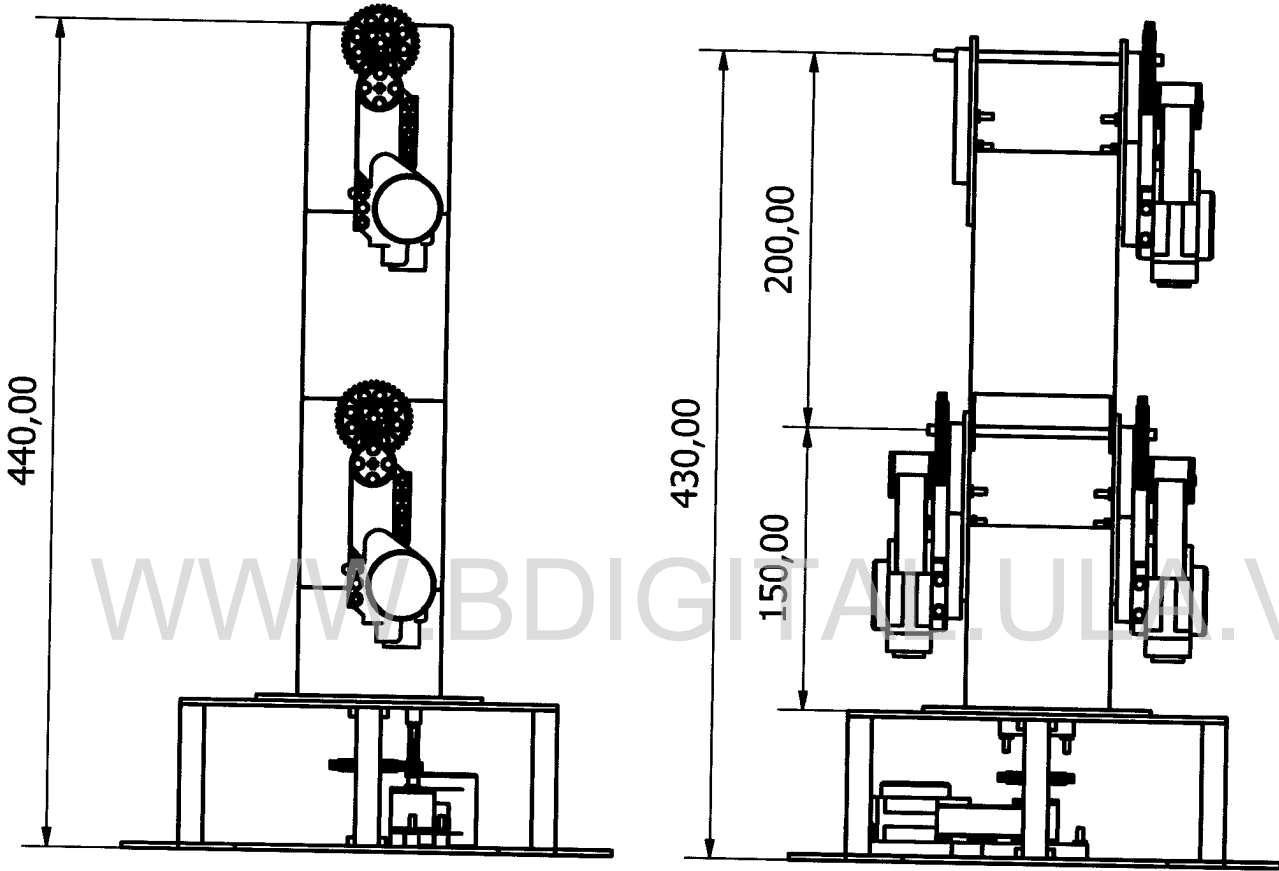
Diseñado por: Luis Andueza	Especialización y Maestría en Ingeniería de Control y Automatización		Fecha: 22/03/2008	Unidades: (mm)
 UNIVERSIDAD DE LOS ANDES VENEZUELA		Brazo Manipulador		
Número de Parte Motor base		Escala Indicada	Hoja 25 / 28	

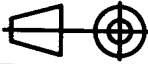

(1 : 4)



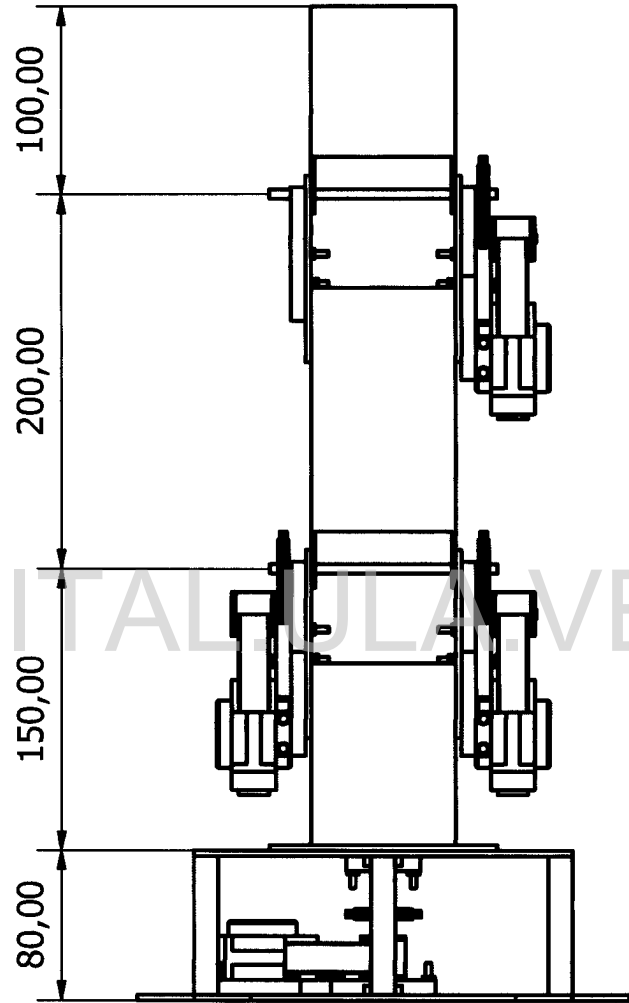
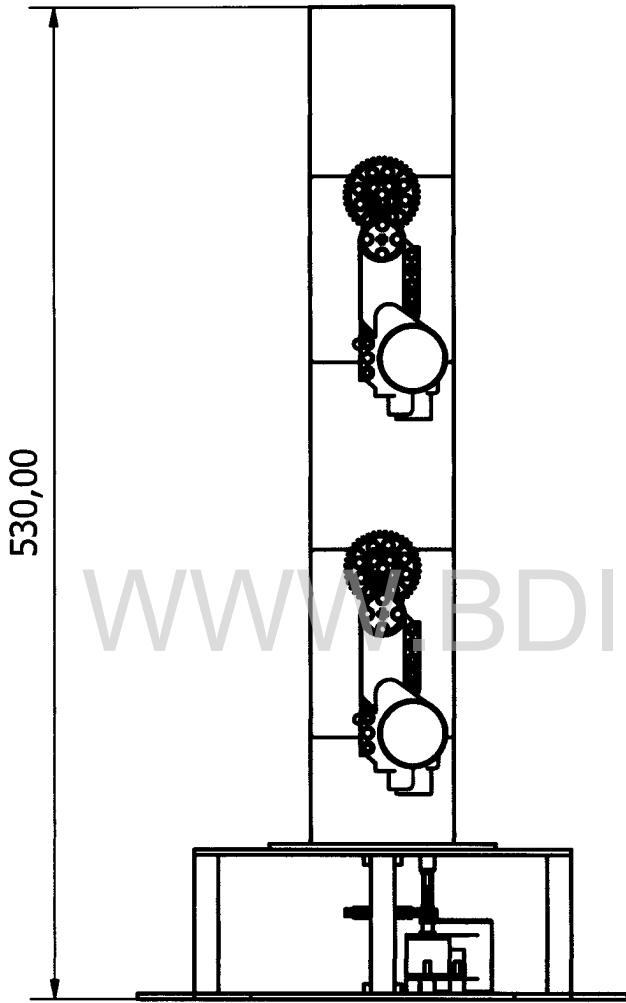
Diseñado por: Luis Andueza	Especialización y Maestría en Ingeniería de Control y Automatización		Fecha: 22/03/2008	Unidades: (mm)
 UNIVERSIDAD DE LOS ANDES VENEZUELA		Brazo Manipulador		
Numero de Parte: Brazo manipulador			Escala Indicada	Hoja 26 / 28



(1 : 4)



Diseñado por: Luis Andueza	Especialización y Maestría en Ingeniería de Control y Automatización		Fecha: 22/03/2008	Unidades: (mm)
 UNIVERSIDAD DE LOS ANDES VENEZUELA	Brazo Manipulador			
Número de Parte: Brazo manipulador		Escala Indicada	Hoja 27 / 28	

(1 : 4)



Diseñado por: Luis Andueza	Especialización y Maestría en Ingeniería de Control y Automatización		Fecha: 22/03/2008	Unidades: (mm)
 UNIVERSIDAD DE LOS ANDES VENEZUELA	Brazo Manipulador			
Número de Hoja: Brazo manipulador			Escala Indicada	Hoja 28 / 28