UACM
Universidad Autónoma
de la Ciudad de México
Nada humano me es ajeno

COLEGIO DE CIENCIA Y TECNOLOGÍA

LICENCIATURA EN INGENIERÍA EN SISTEMAS
ELECTRÓNICOS INDUSTRIALES

Conversión y control CNC en robots para movimiento en el espacio 3D

TESIS

PARA OPTAR POR EL TÍTULO DE
LICENCIADO EN INGENIERÍA EN SISTEMAS
ELECTRÓNICOS INDUSTRIALES

P R E S E N T A :
DANIEL HERNANDEZ CRUZ

D I R E C T O R
DR. JUAN CARLOS MARTÍNEZ ROSAS

Ciudad de México, marzo 2020

UACM
Universidad Autónoma de la Ciudad de México
Nada humano me es ajeno

CA Coordinación Académica

SISTEMA BIBLIOTECARIO DE INFORMACIÓN Y DOCUMENTACIÓN



UNIVERSIDAD AUTÓNOMA DE LA CIUDAD DE MÉXICO
COORDINACIÓN ACADÉMICA

RESTRICCIONES DE USO PARA LAS TESIS DIGITALES

DERECHOS RESERVADOS©

La presente obra y cada uno de sus elementos está protegido por la Ley Federal del Derecho de Autor; por la Ley de la Universidad Autónoma de la Ciudad de México, así como lo dispuesto por el Estatuto General Orgánico de la Universidad Autónoma de la Ciudad de México; del mismo modo por lo establecido en el Acuerdo por el cual se aprueba la Norma mediante la que se Modifican, Adicionan y Derogan Diversas Disposiciones del Estatuto Orgánico de la Universidad de la Ciudad de México, aprobado por el Consejo de Gobierno el 29 de enero de 2002, con el objeto de definir las atribuciones de las diferentes unidades que forman la estructura de la Universidad Autónoma de la Ciudad de México como organismo público autónomo y lo establecido en el Reglamento de Titulación de la Universidad Autónoma de la Ciudad de México.

Por lo que el uso de su contenido, así como cada una de las partes que lo integran y que están bajo la tutela de la Ley Federal de Derecho de Autor, obliga a quien haga uso de la presente obra a considerar que solo lo realizará si es para fines educativos, académicos, de investigación o informativos y se compromete a citar esta fuente, así como a su autor ó autores. Por lo tanto, queda prohibida su reproducción total o parcial y cualquier uso diferente a los ya mencionados, los cuales serán reclamados por el titular de los derechos y sancionados conforme a la legislación aplicable.

Dedicatoria

A mis padres, Margarita Cruz y Juventino Hernandez, por el apoyo económico y moral brindado durante el proceso de mi formación profesional.

Agradecimientos

Al Dr. Juan Carlos Martínez, que acepto ser mi director de tesis. Y por la gran enseñanza que me brindo en el área de robótica.

A mis lectores Dr. Marcos Ángel González, Dr. Cuauhtémoc Guerrero y al Ing. Agustín Ortiz, por el tiempo y conocimiento brindado.

A mis amigos, (UACM) Miguel, Oscar, Marcos y Edgar y (Cia Robotics) Isaac, Oscar y Alejandro, por su amistad y consejos brindados.

Quisiera reconocer a la empresa Cia Robotics por haberme permitido hacer uso de sus diferentes maquinarias y prestarme su robot SCORBOT ER-III para el desarrollo de esta tesis. También a sus colaboradores que fueron pieza clave para poner en operación el brazo robótico ABB IRB6600.

A la empresa IJ-Robotics por el mantenimiento de las licencias de los softwares MasterCAM y RobotMaster y la ayuda de su personal para aprender a utilizar ambos softwares.

Por último agradecer a la Uiversidad Autónoma de la Ciudad de México por el valioso apoyo económico para la impresión de este trabajo.

Es gracias a ustedes que es posible el presente trabajo.

Resumen

La realización de este trabajo recepcional tiene como objetivo implementar un sistema CNC mediante un brazo robótico didáctico (robot académico) con recursos limitados y realizar tareas que sólo se pueden hacer con dispositivos y softwares especializados de elevado costo. Para ello, se diseñará la parte de control y manipulación de un brazo robótico académico, el cuál sólo cuenta con su estructura mecánica, sensores, actuadores y cables de señales; a través de un microprocesador, amplificadores de potencia y una PC.

Se pretende no sólo realizar cálculos para determinar las ecuaciones cinemáticas y dinámicas del robot, sino implementar estos cálculos en el microcontrolador y añadirle una ley de control con compensación de gravedad en el Entorno de Desarrollo Integrado de Arduino (IDE, por sus siglas en inglés), con sintonización de ganancias y análisis los resultados.

Para la parte del diseño de piezas 3D se hará uso de los softwares MasterCAM 2019, en la de generación de código G el software SheetCAM, para la generación de programas para el robot se hará uso de Mach3 y RobotMaster V. 6.6 y para la transferencia del programa al controlador del robot académico DENSO VP-6442G el software WinCaps V. 3.53.0. y para el robot SCORBOT ER-III el IDE de Arduino.

Se presenta una comparativa entre las limitaciones, alcances, costos y beneficios de estos dos sistemas.

Abstract

The goal of this receptional work to implement a CNC system using a didactic robotic arm (academic robot) with limited resources and perform tasks that can only be done with specialized devices and software of high cost. For this, the control and manipulation part of an academic robotic arm will be designed, which only has its mechanical structure, sensors, actuators and signal cables; through a microprocessor, power amplifiers and a PC.

It is intended not only to perform calculations to determine the kinematic and dynamic equations of the robot, but to implement these calculations in the microcontroller and add a control law with gravity compensation in the Integrated Development Environment of Arduino (IDE), with profit tuning and analysis of results.

For the part of the 3D parts design, the MasterCAM 2019 software will be used, in the G code generation the SheetCAM software, for the generation of codes for the robot, use of Mach3 and Robot Master V. 6.6 will be used and for the transfer of the code to the academic DENSO VP-6442G robot controller the WinCaps V. 3.53.0 software. and for the SCORBOT ER-III robot the Arduino IDE.

A comparison between the limitations, scope, costs and benefits of these two systems is presented.

Índice general

Índice de figuras xv

Índice de tablas xix

1. Introducción 1
- 1.1. Presentación . . . 1
- 1.2. Objetivo general . . . 3
- 1.3. Objetivo específico . . . 3
- 1.4. Planteamiento del problema . . . 4
- 1.5. Metodología . . . 6
- 1.6. Estructura de la tesis . . . 7

2. Marco teórico 9
- 2.1. Características de los brazos robóticos . . . 9
  - 2.1.1. Estructuras de brazos robóticos . . . 9
  - 2.1.2. Grados de libertad . . . 10
  - 2.1.3. Espacio de trabajo . . . 11
  - 2.1.4. Sensores . . . 11
  - 2.1.5. Actuadores . . . 13
  - 2.1.6. Sistema de transmisión . . . 15
  - 2.1.7. Trayectoria de un robot . . . 17
  - 2.1.8. Efector final . . . 17
  - 2.1.9. Programación de robots . . . 17
- 2.2. Sistemas CAD/CAM automatizados . . . 18
- 2.3. Sistema CNC . . . 19
- 2.4. Código G y M . . . 19

3. Análisis cinemático y dinámico del robot SCORBOT ER-III con 3 DOF 21
- 3.1. Cinemática directa . . . 21
- 3.2. Parámetros Denavit Hartenberg (PDH) . . . 21

3.3. Algoritmo para determinar la matriz de transformación homogénea, A_n . . . 23
3.4. Cinemática inversa . . . 27
3.5. Dinámica de un robot . . . 31
3.5.1. Modelo dinámico de un robot . . . 31
3.6. Sistema de control para un brazo robótico . . . 39
3.6.1. Control de posición PD con compensación de gravedad . . . 40

4. Análisis de Resultados **43**
4.1. Desarrollo . . . 43
4.1.1. Implementación de la ley de control y conversión del robot SCORBOT ER-III a máquina CNC . . . 43
4.1.2. Conversión del robot DENSO VP-6442G a máquina CNC . . . 49
4.2. Resultados . . . 53
4.2.1. Resultados de la conversión del robot SCORBOT ER-III como máquina CNC . . . 53
4.2.2. Resultados de la conversión del robot DENSO VP-6242G como máquina CNC . . . 53
4.3. Comparativa . . . 56

5. Conclusiones y trabajos futuros **59**
5.1. Conclusiones . . . 59
5.2. Mejoras y trabajos futuros . . . 60

A. Datos técnicos, robot DENSO VP-6442G, SCORBOT ER-III **63**

B. Programas **69**
B.1. Programa en lenguaje C para el microcontrolador ATMEGA 2560 69
B.2. Programas en lenguaje G . . . 76

C. Circuitos electrónicos **77**
C.1. Driver para motor puente H, BTS7960 . . . 77
C.2. IC Schmitt Trigger modelo 74ls14n . . . 78
C.3. Convertidor USB a TTL FT232RL . . . 79
C.4. Arduino Mega 2560 . . . 80
C.5. Shield para el proyecto . . . 82

D. Uso del software **85**
D.1. MasterCAM . . . 85
D.2. RobotMaster . . . 97
D.3. WinCaps iii . . . 104

E. Licencias **109**

Bibliografía **111**

Índice de figuras

1.1. Aplicaciones de robots industriales. 2
1.2. Ventas mundiales anuales de Robots industriales por región. . . . 5
1.3. Venta anual de robot industriales por tipo de industria. 5
1.4. Venta anual de robot industriales de los 15 grandes mercados en 2018. 6

2.1. Configuraciones de robots manipuladores. 10
2.2. Área de trabajo, robot de 6 DOF marca ABB modelo IRB 6600-175. *En "Especificaciones del producto, robot articulado IRB 6600", 2004, Copyright [2004] por ABB.* 11
2.3. Encoder tipo absoluto. *En "Robotics", (p. 54) Bajd et al. [2010]. Springer.* 12
2.4. Encoder tipo incremental. *En "Robotics", (p. 55) Bajd et al. [2010]. Springer.* 13
2.5. Actuadores y sensores eléctricos robot ABB. *A. Miller, (2011) " New ABB Robotics Motor and Gear Unit Packages", recuperado de http://www.zycon.com* 14
2.6. Transmisión armónica. *I. Tamboli, (2019) "Harmonic Drive Gear Market Analysis by Recent Developments and Growth Outlook 2019", recuperado de http://www.galusaustralis.com* 16

3.1. Parámetros Denavit-Hartenberg 22
3.2. Representacion de los PDH Scorbot ER-III 25
3.3. Multiples soluciones de ángulos en un robot 28
3.4. Solución cinemática inversa codo abajo 29
3.5. Solución cinemática inversa codo arriba 30
3.6. Inercia en un eslabón 31
3.7. Representación dinámica para el robot SCORBOT ER-III 33
3.8. Sistema de control general para un brazo robótico 39
3.9. Control PD con compensación de gravedad 40

4.1. Diseño del proyecto 44
4.2. Archivo para transmitir posiciones X, Y, Z de Mach3 al IDE de Arduino 45
4.3. Configuración del protócolo Modbus en Mach3 46
4.4. Prueba del protócolo Modbus en Mach3 47
4.5. Conexiones entre los modulos de adquisición de datos y potencia para el robot académico SCORBOT ER-III 48
4.6. Base para el efector final del robot DENSO VP-6442G 49
4.7. Modificación de parámetros de cortadores en Mastercam 50
4.8. Pieza "logo UACM", en MasterCAM para el robot DENSO VP-6442G 51
4.9. Simulación de mecanizado de la pieza "logo UACM", en Robotmaster 52
4.10. Pieza "logo UACM", realizado en acrílico con el robot DENSO VP-6442G como CNC 54
4.11. Pieza "Engrane cónico", realizada en MDF con el robot DENSO VP-6442G como CNC 55

A.1. Dimensiones para efector final del robot DENSO VP-6442G . . . 64
A.2. Área de trabajo del robot DENSO VP-6442G. Vista lateral 64
A.3. Características generales del robot DENSO VP-6242G 65
A.4. Dimensiones de la base del robot DENSO VP-6442G 66
A.5. Área de trabajo del robot DENSO VP-6442G. Vista superior . . . 66
A.6. Características generales del robot SCORBOT ER-III 67

C.1. Modulo driver BTS7960 77
C.2. Características generales en DC del IC 74ls14n 78
C.3. Diagrama esquemático y pinout del IC 74ls14n 78
C.4. Conformación del pulso del IC 74ls14n 79
C.5. Modulo USB a TTL FT232RL 79
C.6. Arduino Mega 2560 80
C.7. Pinout Arduino Mega 2560 81
C.8. Diseño de la PCB en Eagle 83
C.9. Vista inferior y superior de la PCB 84

D.1. Entorno MasterCAM 85
D.2. Visualización pieza 3D en MasterCAM 86
D.3. Selección de máquina en MasterCAM 86
D.4. Maquinados en 2D, 3D y multi ejes en MasterCAM 87
D.5. Tipos de corte en MasterCAM 87
D.6. Caja envolvente en MasterCAM 88
D.7. Material bruto de la pieza de trabajo en MasterCAM 88

D.8. Tipo de herramienta de corte en MasterCAM 89
D.9. Ajustes de maquinado en MasterCAM 90
D.10.Tipo de herramienta de corte en MasterCAM 91
D.11.Ajustes de porta herramienta en MasterCAM 91
D.12.Parámetros de corte en MasterCAM 92
D.13.Parámetros entrada/salida de corte en MasterCAM 92
D.14.Visualización del maquinado 3D en MasterCAM 93
D.15.Selección de trayectoria multi eje en MasterCAM 93
D.16.Patrón de corte multi eje en MasterCAM 94
D.17.Control del eje de herramienta de corte en MasterCAM 94
D.18.Vinculación de corte en MasterCAM 95
D.19.Visualización del maquinado multi eje en MasterCAM 95
D.20.Ventana de simulación en MasterCAM 96
D.21.Pestaña para cambiar de MasterCAM a RobotMaster 96
D.22.Entorno RobotMaster 97
D.23.Ajustes de simulación en RobotMaster 97
D.24.Selección del robot en RobotMaster 98
D.25.Selección del efector final en RobotMaster 98
D.26.Configuración de la pieza de trabajo en RobotMaster 99
D.27.Ajuste de colisiones en RobotMaster 99
D.28.Robot, efector final y pieza de trabajo en RobotMaster 100
D.29.Optimización de trayectorias en RobotMaster 100
D.30.Visualización de trayectorias de corte en RobotMaster 101
D.31.Simulación de las trayectorias de corte en RobotMaster 101
D.32.Generación de programa de robot en RobotMaster 102
D.33.Movimiento de cada articulación en RobotMaster 102
D.34.Postprocesador en MasterCAM 103
D.35.Archivos generados en Code Expert 103
D.36.Creación de un nuevo proyecto en Wincaps iii y dirección del archivo104
D.37.Selección del tipo de robot en Wincaps iii 105
D.38.Selección y especificaciones del robot en Wincaps iii 105
D.39.Tipo de conexión y dirección en Wincaps iii 106
D.40.Especificaciones del proyecto en Wincaps iii 107
D.41.Entorno de trabajo en Wincaps iii 107

E.1. Licencias de prueba 109
E.2. Licencias de prueba 110

Índice de tablas

1.1. Ventas anuales de robots 4

3.1. Parámetros obtenidos para el robot SCORBOT ER-III. 25
3.2. Constantes dinámicas para el robot SCORBOT ER-III. 38

4.1. Ganancias para el control de posición del robot SCORBOT ER-III. 48
4.2. Parámetros de cada herramienta de corte. 50
4.3. Medidas de corte de la pieza realizada "logo UACM". 54
4.4. Medidas de corte de la pieza realizada "engrane cónico". 56
4.5. Presición de las piezas realizadas con el robot DENSO VP-6442G. 56
4.6. Comparativa entre los diferentes robots y máquinas CNC. 57

C.1. Pines para serial Arduino Mega 2560 82
C.2. Pines para interrupciones Arduino Mega 2560 82

Capítulo 1

Introducción

1.1. Presentación

Durante la historia de existencia de la humanidad se sabe que el ser humano ha buscado mejorar sus condiciones de vida. Para ello ha desarrollado el uso de herramientas como la palanca, la rueda, el tornillo, la polea, la cuña; que le permite realizar tareas o actividades de una mejor manera. Hoy en día, con la existencia de máquinas, el ser humano ha podido realizar tareas o procesos que presentaban limitaciones para poder llevarlas a cabo. Estas máquinas pueden ser sistemas automatizados, equipos de Manufactura Asistida por Computadora (CAM, por sus siglas en inglés), equipos robot, entre otros. Para el caso de los robots, la demanda de ellos ha sido muy grande en países como Japón, Estados Unidos de América, Alemania entre otros países europeos y asiáticos.

La palabra robot proviene del vocablo checo "Robota" que significa trabajo arduo, repetitivo y monótono. La Asociación de Industrias de Robótica (RIA, por siglas en inglés) define el término robot industrial como un manipulador multifuncional reprogramable, diseñado para mover materiales, piezas, herramientas o dispositivos especiales, según trayectorias variables programadas para realizar tareas diversas. Por otra parte, la manipulación se define como la acción de sujetar un objeto y cambiar la posición y orientación de ésta en el espacio 3D. También la RIA menciona que los usos comunes de los robots industriales en los últimos años, son las operaciones de soldadura, pintura, manipulación de materiales, ensamblaje y acabado por aspersión.

En la actualidad podemos encontrar robots en casi cualquier lugar: hospitales, almacenes, industrias manufactureras, industrias ensambladoras, laboratorios de investigación, la aeronáutica, entre otros como se muestra en la Figura 1.1; lo que

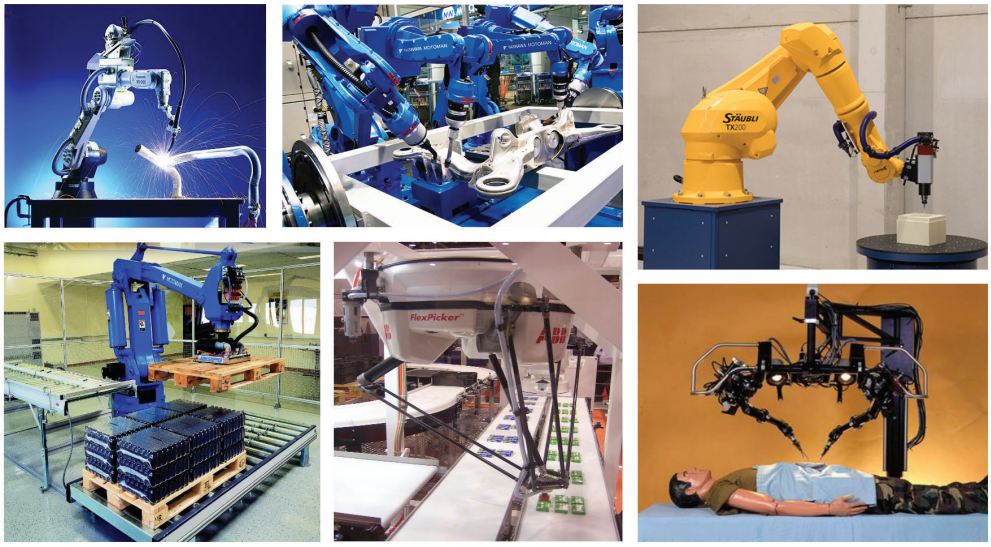


Figura 1.1: Aplicaciones de robots industriales.

ha empujado el reemplazo de trabajadores por máquinas. Entre los beneficios están [Hernandez and Vizán (10)]:

- Mejor calidad de acabado, ya que los robots cuentan con un alto grado de precisión y repetibilidad.
- Reducción de espacio.
- Trabajo en cualquier ambiente: en comparación con un humano el robot puede realizar grandes esfuerzos, trabajar en ambientes peligrosos sin que sufra daño alguno.
- Reducción de pérdidas: tanto de tiempo como de desperdicios.

Estos beneficios incrementan la productividad de las industrias, lo cual hace que puedan ser más competitivos a nivel internacional.

En el mercado podemos encontrar diferentes empresas dedicadas a la fabricación de robots industriales, así como diferentes modelos que tienen características especiales. Entre las empresas que desarrollan robots están ABB, KUKA, FANUC, DENSO, MITSUBISHI, MOTOMAN, SCORBOT, KAWASAKI. La elección de algún modelo va a depender del uso que se le dé al robot, así como las capacidades de éste.

El desarrollo e investigación de nuevas tecnologías de brazos robóticos ha logrado que estos ofrezcan mayor rendimiento, flexibilidad, rigidez y capacidad de carga, además las aplicaciones cada vez son más extensas. En los últimos años se ha observado grandes mejoras en la parte de comunicaciones, también en la cooperación y sincronización de diversos robots.

El futuro de la robótica industrial va encaminado a programación por aprendizaje, manipuladores de más de 6 grados de libertad (DOF, por sus siglas en inglés), control por visión artificial, comunicaciones inalámbricas.

1.2. Objetivo general

Este trabajo tiene por objetivo desarrollar una aplicación en el área de ingeniería robótica, que tiene por finalidad convertir el modo de trabajo para el cual fueron desarrollados los robots, el cual consiste en transformar su forma de trabajo de operación punto-punto a operación Control Numérico Computarizado (CNC, por sus siglas en inglés) para seguimiento de trayectorias, mediante el uso de lenguaje G. La aplicación es un sistema de mecanizado CNC para desbaste mediante diferentes brazos robóticos académicos: uno especializado con su sistema de control y el otro sin su sistema de control. Posteriormente, debe realizarse una comparativa de costo-beneficio y limitaciones entre los diferentes brazos robóticos y una máquina CNC.

1.3. Objetivo específico

Se contempla realizar el sistema de mecanizado CNC en tiempo real mediante el brazo robótico que no cuenta con sistema de control, sino que sólo cuenta con sensores, actuadores y la estructura mecánica. Para ello, al robot se le tiene que realizar toda la parte de modelado, control, adquisición de señales y manipulación de datos. Por último, los cálculos obtenidos no sólo se deben simular, sino que se deben implementar en algún sistema embebido de bajo costo, pero con buenas cualidades y la parte de circuitos debe realizarse con dispositivos utilizados durante la carrera.

1.4. Planteamiento del problema

El área de robótica industrial ha logrado que varios trabajos peligrosos realizados anteriormente por humanos, se convierta en una actividad segura y precisa. Además, ya puede ser considerado como uno de los principales motores de competitividad y flexibilidad de las industrias manufactureras, lo que se traduce en progreso tecnológico. Sin embargo, el costo de instalación, capacitación y operación de este tipo de maquinaria es muy elevado, y es por ello que en México se adoptan este tipo de tecnologías de una manera más reducida y lenta que en otros países más desarrollados.

Country	2016	2017	2018*	2019*	2020*	2021*	2018/ 2017	CAGR 2019 - 2021
America	41,295	46,118	44,300	48,900	55,600	63,500	-4%	13%
North America	39,671	43,529	43,000	47,500	54,000	61,500	-1%	13%
- United States	31,404	33,192	35,000	37,500	41,000	46,000	5%	10%
- Canada	2,334	4,003	3,500	4,000	5,500	6,500	-13%	23%
- Mexico	5,933	6,334	4,500	6,000	7,500	9,000	-29%	26%
Brazil	1,207	961	900	900	1,000	1,200	-6%	10%
Rest of South America	394	300	400	500	600	800	33%	26%
America unspecified**	23	1,328						
Asia/Australia	190,542	261,826	298,150	351,250	405,400	462,600	14%	16%
China	87,000	137,920	165,000	210,000	250,000	290,000	20%	21%
India	2,627	3,412	4,500	5,000	6,000	7,500	32%	19%
Japan	38,586	45,566	54,000	56,000	59,000	64,000	19%	6%
Republic of Korea	41,373	39,732	41,000	42,000	44,500	46,000	3%	4%
Taiwan, Province of China	7,569	10,904	13,000	14,000	17,000	20,000	19%	15%
Thailand	2,646	3,386	4,000	5,000	6,000	7,000	18%	21%
Vietnam	1,618	8,252	2,500	3,000	4,500	7,000	-70%	41%
other Asia/Australia	9,123	12,654	14,150	16,250	18,400	21,100	12%	14%
Europe	56,078	66,259	70,950	75,250	82,500	93,600	7%	10%
Central/Eastern Europe	7,758	10,538	13,500	16,500	19,750	24,300	28%	22%
France	4,232	4,897	5,200	5,600	6,000	6,500	6%	8%
Germany	20,074	21,404	22,500	23,500	25,000	26,000	5%	5%
Italy	6,465	7,713	9,000	9,000	9,500	10,500	17%	5%
Spain	3,919	4,180	4,700	4,600	5,100	6,500	12%	11%
United Kingdom	1,787	2,334	2,400	2,200	2,300	2,600	3%	3%
Rest of Europe	11,706	12,133	12,850	13,300	14,350	16,600	6%	9%
Europe unspecified**	137	3,060	800	550	500	600	-74%	-9%
Africa	879	451	500	600	700	800	11%	17%
not specified by countries**	5,553	6,681	7,100	8,000	8,800	9,500	6%	10%
TOTAL	294,347	381,335	421,000	484,000	553,000	630,000	10%	14%

Tabla 1.1: Ventas anuales de robots - Número de unidades estimadas de ventas anuales de robots industriales por ciudades.

La International Federation of Robotics reporta que para el año 2018 la ventas de robots industriales incrementaron en 19 % del año 2013 al 2018. Este crecimiento se debió a las industrias del metal, industrias automotrices e industrias eléctricas/eletrónicas como se muestra en la Figura 1.3. En el año 2017 el aumento anual promedio de venta de robots industriales fue de aproximadamente 400,000 unidades. En la Figura 1.2 se puede observar a Asia como el continente con mayores ventas de robots en el mundo, para el año 2018 logró vender aproximadamente 283,080 unidades. Le sigue Europa con casi 75,560 unidades y América con 55,212 unidades. En la Figura 1.4 se observa que México para el año 2018 vendió 5,700 unidades y de la Tabla 1.1 se esperá que para el año 2021 venda aproximadamente 9,000 unidades teniendo un crecimiento de 26 %.

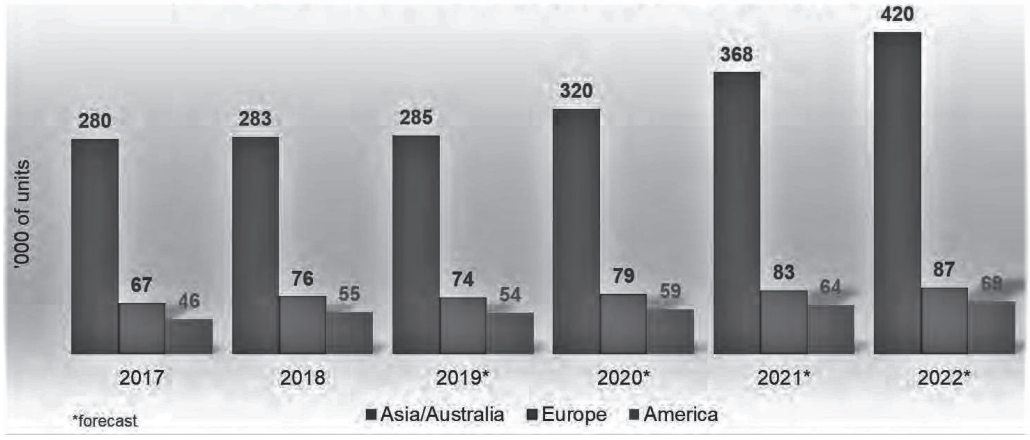


Figura 1.2: Ventas mundiales anuales de Robots industriales por región.

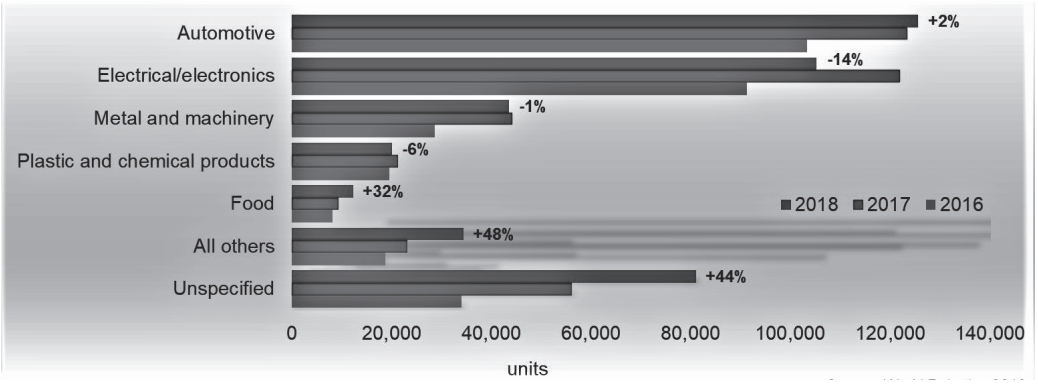


Figura 1.3: Venta anual de robot industriales por tipo de industria.

Entonces surge la pregunta: ¿Por qué en México no existe una empresa mexicana que desarrolle tecnología para robots industriales?, siendo que el mismo país es un gran mercado para su venta. Actualmente las ventas de robots industriales realizadas en México han sido de empresas extranjeras.

Basado en un estudio de "The Competitive Intelligence Unit", México ocupa el puesto 23 de 25 países evaluados en políticas educativas, innovación y políticas laborales en la adopción de robótica avanzada en las industrias. Este estudio también menciona que México es el principal importador de robots industriales de América latina. Sin embargo como se mencionó anteriormente, México avanza lento en la parte de innovación y educación en el área de robótica, ya que sólo se le destina el 0.55 % del PIB a la investigación y desarrolo de nuevas tecnologías.

Es por todo esto que resulta viable y además contribuye en la generación de nuevas tecnologías la realización de este proyecto de tesis.

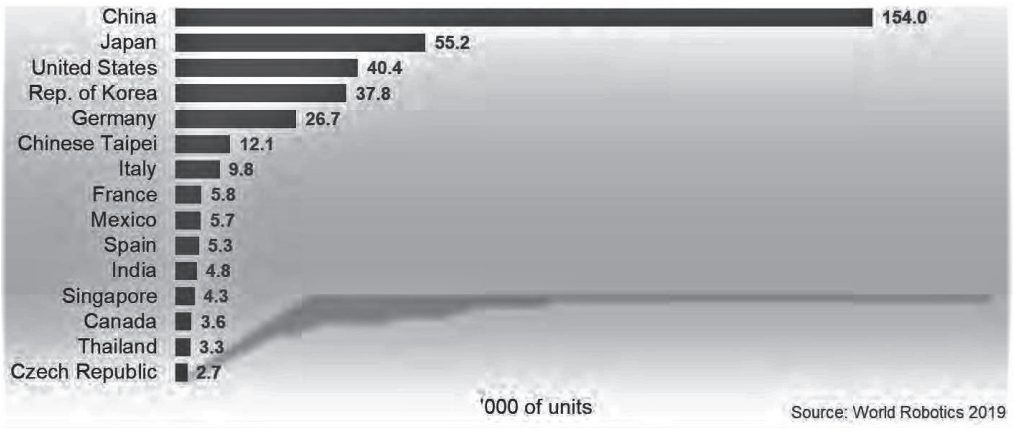


Figura 1.4: Venta anual de robot industriales de los 15 grandes mercados en 2018.

1.5. Metodología

Para el desarrollo de esta aplicación se hará uso del equipo con el que cuenta el laboratorio de robótica de la UACM plantel San Lorenzo y equipo de la empresa CIA Robotics. Este equipo es el siguiente:

- Robot SCORBOT ER-III.

- Robot DENSO VP-6442G.
- Robot ABB IRB-6600.
- Computadora con tarjeta gráfica de 6GB.
- Microcontrolador Arduino Mega 2560, módulos amplificadores de potencia.
- Licencias de los softwares MasterCAM, RobotMaster, los softwares libres Wincaps iii, IDE de Arduino, Mach3, SheetCAM TNG Development.

En el robot académico SCORBOT ER-III se diseñará y construirá la etapa de adquisición y manipulación de las señales, se implementará la etapa de control en el microprocesador del Arduino Mega 2560 y se enlazará estas estapas con la de amplificación de potencia eléctrica. Para el robot DENSO VP-6442G sólo se manipulará el programa de robot.

Cada uno de estos robots necesita un efector final, para ello se construirán diferentes estructuras para poder montarlos en ellos. Las estructuras se realizarán en la empresa CIA Robotics, ya que la empresa cuenta con diferente maquinaria industrial.

1.6. Estructura de la tesis

Este trabajo está dividido en 5 capítulos. En el segundo capítulo trata un marco teórico de las caracteristicas que debe tener un brazo robótico, asi como un sistema CNC. El tercer capítulo contiene preliminares matemáticos para el desarrollo de cálculos necesarios para la cinemática, dinámica y ley de control del robot académico SCORBOT ER-III. En el cuarto capítulo se muestra el desarrollo del proyecto, tanto circuitos electrónicos, programas, la parte de control, así como el análisis de los resultados obtenidos. Por último, en el quinto capítulo se mencionarán conclusiones, aportaciones y trabajos futuros respecto al proyecto de tesis.

Se incluyen 5 anexos. El anexo A incluye la hoja de datos técnica de los diferentes robots. En el anexo B se muestra los programas utilizados en los diferentes softwares. En el anexo C se muestran las hojas de datos de los diferentes circuitos electrónicos, así como el desarrollo de la Placa de Circuito Impreso (PCB, por sus siglas en inglés). En el anexo D se muestra el uso y mantenimiento de cada uno de los diferentes softwares. En el anexo E se muestra las licencias que fueron

adquiridas.

Comentario 1.6.1 *Los cálculos realizados en el tercer capítulo sólo son para el brazo robótico SCORBOT ER-III ya que este es el que no cuenta con sistema de control. El robot DENSO VP-6442G ya cuenta con su sistema de control. En el cuarto capítulo se hace la conversión de ambos robots en un sistema CNC.*

Capítulo 2

Marco teórico

2.1. Características de los brazos robóticos

2.1.1. Estructuras de brazos robóticos

Las configuraciones más comunes de robots articulados que se encuentran en el mercados son 5: antropomórfico, cartesiano, cilíndrico, SCARA y polar. Estas se muestran en la Figura (2.1).

- Robots de coordenadas cartesianas (TTT). Son sólo movimientos lineales, el espacio de trabajo del efector final es de forma prisma, se emplean mayormente para manejo y montaje de materiales.
- Robots de cordenadas cilíndricas (RTT). Se tienen dos movimientos lineales y uno rotacional, se emplean principalmente para transportar objetos incluso de grandes dimensiones.
- Robots de coordenadas polares (RRT). Se tienen dos movimientos rotacionales y uno lineal, son principalmente empleados para mecanizado.
- Robots SCARA (RRT). Se tienen dos movimientos rotacionales y uno lineal, el espacio de trabajo del efector final es de forma cilíndrica, son los más adecuados para la manipulación de objetos pequeños.
- robot antropomórfico (RRR). Se tienen sólo movimientos rotacionales, el espacio de trabajo del efector final tiene una forma esférica, las aplicaciones industriales son muchas.

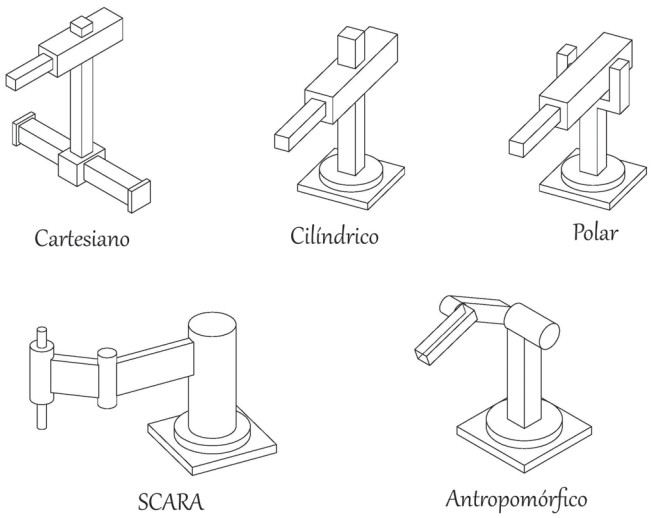


Figura 2.1: Configuraciones de robots manipuladores.

2.1.2. Grados de libertad

El grado de libertad (DOF, por sus siglas en inglés) se define considerando una partícula de masa infinitesimal, como el número de coordenadas independientes que son necesarias para la descripción completa de la posición de una partícula de masa [Bajd et al. (5)].

Los grados de libertad en robótica hacen referencia a los parámetros que determinan tanto la posición como la orientación del efector final en un manipulador. Otra forma de definir los grados de libertad es observar los movimientos que puede realizar el brazo robótico (movilidad), la movilidad se define como el número de variables independientes que se especifican para ubicar las articulaciones del robot respecto a otra articulación [Barrientos et al. (6)]. Se dice que un robot con 6 DOF cubre un espacio de trabajo esférico, a su vez el robot puede moverse sobre cualquiera de sus tres ejes ortonormales y puede rotar angularmente con respecto a cada uno de ellos; si se tiene un robot con mayor DOF se dice que el robot es redundante.

2.1.3. Espacio de trabajo

El área de trabajo es el volumen espacial al que puede llegar el extremo del robot como se muestra en la Figura 2.2. Este espacio de trabajo depende directamente de las especificaciones geométricas de los eslabones, los DOF, así como de las limitaciones de ángulo de giro por parte del sistema de control. Generalmente en la información técnica de cada robot se específica su área de trabajo, sin embargo, que el extremo pueda llegar a un punto en el espacio no necesariamente significa que pueda ubicarse en cualquier orientación.

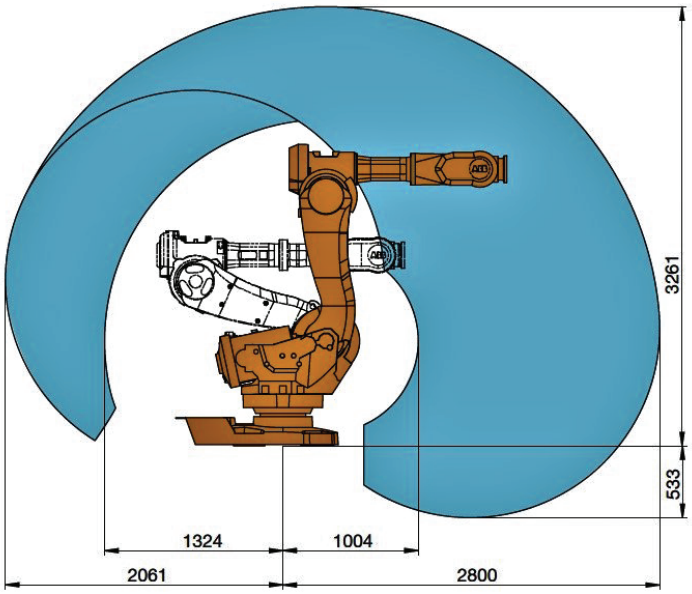


Figura 2.2: Área de trabajo, robot de 6 DOF marca ABB modelo IRB 6600-175. *En "Especificaciones del producto, robot articulado IRB 6600", 2004, Copyright [2004] por ABB.*

2.1.4. Sensores

En un manipulador es esencial contar con sensores de tipo propioceptivos (percepción del estado interno del robot) y exteroceptivos (percepción de los aspectos externos al robot). Un sensor es un dispositivo que mide alguna magnitud física, para después convertirla en una señal eléctrica (transductor), adicionalmente puede contar con un acondicionador de señal para obtener una señal adecuada.

Los sensores que usualmente se encuentran en los manipuladores para medir movimientos son el decodificador (encoder en inglés).

Dentro de los encoder encontramos el óptico, que se basa en un emisor de luz con un diodo, un receptor de luz que lo convierte en corriente eléctrica, ya sea un fotodiodo o un fototransistor y un disco giratorio con ranuras que permiten o bloquean el paso de la luz. A su vez el encoder óptico se divide en absoluto e incremental [Bajd et al. (5)].

- Encoder absoluto. Es un sensor que entraga señales digitales como se observa en la Figura 2.3, con el cual se puede medir la posición angular absoluta una vez desarrollada alguna formula, la resolución del encoder va a depender del número de bits que tenga grabado el disco giratorio, por ende, se van a requerir el mismo número de emisores y receptores. En este tipo de encoder no solo se puede medir la posición angular, sino también se puede saber la dirección, si va del estado lógico LSB al MSB por convención su dirección es positiva, en cambio sí va de un estado lógico MSB al LSB su dirección es negativa.

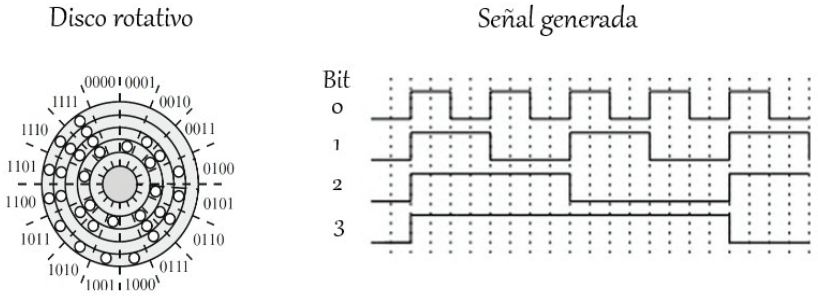


Figura 2.3: Encoder tipo absoluto. *En "Robotics", (p. 54) Bajd et al. [2010]. Springer.*

- Encoder incremental. A diferencia del encoder absoluto, este es más simple y sus dimensiones más pequeñas. Sólo se requiere de un par de emisores y receptores de señal separados a una distancia tal que, cada canal debe estar desfasado 90°, de manera que puedan representar trenes de pulsos con este desface como se observa en la Figura 2.4. La resolución depende de la cantidad de ranuras con las que cuente el disco giratorio. La resolución Δ_ϑ puede calcularse mediante la siguiente formula [Bajd et al. (5)]:

$$\Delta_\vartheta = \frac{2\pi}{n_o} \tag{2.1}$$

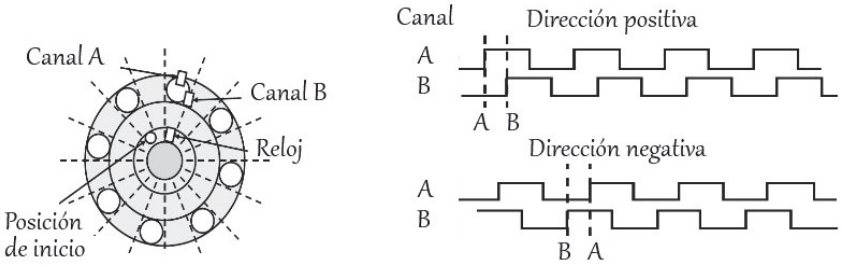


Figura 2.4: Encoder tipo incremental. *En "Robotics", (p. 55) Bajd et al. [2010]. Springer.*

donde n_o es el número de orificos del disco giratorio.

En caso de que se desee aumentar la resolución, no solo basta con contar los flancos de subida sino también los de bajada, con ello la resolución se incrementa por 4.

2.1.5. Actuadores

Los actuadores generan las fuerzas o pares necesarios para animar la estructura mecánica del robot [Baturone (7)]. Entre los sistemas capaces de realizar la acción de actuador para poder mover un eslabón, se encuentran los sistemas neumáticos, hidráulicos o eléctricos. El sistema neumático es muy rápido, pero no es recomendable, debido a que para poder realizar un control preciso ya sea de posición o velocidad, hay problemas para controlar la compresibilidad del aire, siendo este sistema el más complejo. El sistema hidráulico es recomendable para aplicaciones que requieran producir fuerzas o torques de gran magnitud. El sistema eléctrico es el más comúnmente utilizado, ya que tienden a ser más manejables en la parte del control con lo que obtienen mejores resultados en precisión de posición y repetitividad, así como mayor velocidad.

Los robots accionados con sistema eléctrico (motores), requieren de un sistema de control para poder realizar un control de posición y/o control de velocidad. Los

motores eléctricos están compuestos por: armadura, estator, rotor y conmutador. Entre los actuadores eléctricos se encuentran 3 tipos [Barrientos et al. (6)]:



Figura 2.5: Actuadores y sensores eléctricos robot ABB. *A. Miller, (2011) " New ABB Robotics Motor and Gear Unit Packages", recuperado de http://www.zycon.com*

- Motores de corriente directa (DC). Para poder hacer girar el rotor, este debe ubicarse en ángulo recto con respecto a un campo magnético en el estator (inductor), usualmente generado por imanes permanentes o bobinas, el rotor experimenta una fuerza perpendicular a la corriente y a las líneas de fuerza magnética. El rotor transporta las bobinas de la armadura (inducido), colocadas en sentido axial, y una vez que fluye corriente a través de la armadura el rotor comienza a girar. Al variar la corriente de campo o de la armadura, la velocidad en el rotor puede aumentarse o disminuirse. Es de los actuadores eléctricos más utilizados para robots de uso didáctico o industriales de medianas dimensiones. En la Figura 2.5 se muestra los motores de corriente directa que utiliza un brazo robótico de la marca ABB.

- Motores de corriente alterna (AC). El inductor se situa en el rotor y está constituido por imanes permanentes, mientras que el inducido, situado en el estator, está formado por tres devanados iguales decalados 120° eléctricos y se alimentan con un sistema trifásico de tensiones [Barrientos et al. (6)]. La velocidad va a depender de la frecuencia de tensión que alimenta el inducido.

- Motores paso a paso. Este tipo de motor produce movimientos en ángulos fijos, por lo que no requiere de un sistema de control con retroalimenta-

ción, el control se realiza mediante programación de un tren de pulsos. Son utilizados mayormente en pequeños robots didacticos.

2.1.6. Sistema de transmisión

Una máquina contiene generalmente mecanismos diseñados para proporcionar fuerzas significativas y transmitir potencia significativa [Khare et al. (12)]. Los sistemas de transmisión se encargan de transmitir movimiento desde los actuadores hasta las articulaciones, y es necesario en un robot (dada si la unión entre la flecha del motor y el eslabón fuera directa) por el hecho de que se tiene mucha velocidad y poco par. Teniendo algún sistema de transmisión se puede adaptar tanto el par como la velocidad requeridas en cada una de las articulaciones. Los sistemas de transmisión deben de tener un peso reducido, pero debe ser un material de buena calidad, es decir, que sea susceptible al desgaste u oxidación, debe de evitar tener holguras o juego entre dos o más elementos. A continuación se describen los mas usuales: engranes, bandas dentadas, tornillos sinfin y reductores armónicos.

- Engranes. Es un sistema el cual puede hacer un cambio de dirección o velocidad de movimiento, lo que implica un cambio en el torque transmitido. Existen 4 tipos principales de engranajes: engranajes rectos, engranajes de cremallera y piñon, engranajes cónicos y engranajes helicoidales.

  - Engranes rectos. De los más utilizados, consiste en 2 o más engranes colocados de manera paralela a los ejes de rotación, de las mismas o diferentes dimensiones, pero con las mismas dimensiones en los dientes.
  - Engranes piñon y cremallera. Es un sistema compuesto por 2 engranes, uno de tipo circular (piñon) y otro recto (cremallera).
  - Engranes cónicos. Se compone de dos engranes con dientes de forma cónica, colocados en ángulos entre los ejes de rotación. Es de los engranes más ocupados en robótica por sus dimensiones, poco desgaste, transmisión de torque elevados y poco ruido.
  - Engranes helicoidales. Estos pueden ser colocados de forma paralela u orientación cruzada entre los ejes de rotación. Los bordes de los dientes no son paralelos al eje de rotación, pero están colocados en ángulo. Los dientes en ángulo se acoplan de manera que el movimiento entre ellos es más suave y sin ruido a altas velocidades, además soporta altas torsiones.

- Tornillos sinfin. Compuesto por un tornillo y un engrane de manera perpendicular, a medida que el tornillo gira el engrane también gira.
- Bandas dentadas. Generalmente presentan poco desplazamiento, pero son poco durables debido a que debe estar bien tensa, no emiten ruido, además son más complicadas de controlar, esto a causa de que se requiere una alineación precisa entre los ejes del piñon y del actuador.
- Reductores armónicos. El accionamiento armónico se basa en un principio llamado transmisión de movimiento por onda-deformación (strain-wave gearing), un nombre derivado del funcionamiento de su elemento primario de transmisión de par, la flexspiline [Sandin (17)]. Compuesto por tres elementos como se observa en la Figura 2.6. Entre las ventajas de utilizar este sistema de transmisión en un robot está que los ejes de entrada y salida tienen la misma línea central. Tiene excelente precisión de posicionamiento y repetibilidad. Son compactos y ligeros. Y lo mejor es que cuenta con una muy alta precisión de transmisión sin presentar holguras (zero backlash).

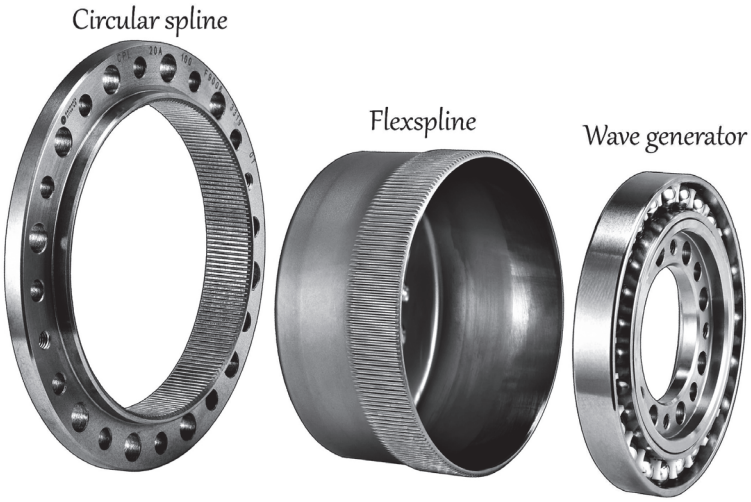


Figura 2.6: Transmisión armónica. *I. Tamboli, (2019) "Harmonic Drive Gear Market Analysis by Recent Developments and Growth Outlook 2019", recuperado de http://www.galusaustralis.com*

2.1.7. Trayectoria de un robot

Los tipos de trayectorias posibles para llegar de un punto a otro son principalmente tres: punto a punto (PTP, por sus siglas en inglés), de trayectoria continua (CP, por sus siglas en inglés) y de trayectoria en arco (ARC, por sus siglas en inglés).

Para el movimiento "PTP"solo se requiere de la posición inicial y de la posición final, la trayectoria entre ambos puntos no será siempre predecible, es decir, existen varias trayectorias aleatorias para llegar al punto, que dependen de la postura del robot. Si se trata de distancias pequeñas entre los dos puntos tal vez sea predecible el movimiento que el robot realizará. Generalmente este tipo de trayectoria es poco utilizada para aplicaciones de mecanizados, soldadura, pintura.

El movimiento "CP"gestiona una interpolación suave y continua a lo largo de una trayectoria deseada, de tal manera que el movimiento sea lo más parecido a una línea recta.

El movimiento "ARC"también hace uso de la interpolación pero en este caso requiere de un tercer punto de referencia, el cual tomará como punto medio para realizar el arco entre los puntos inicial y final.

2.1.8. Efector final

El efector final es la pieza que se encuentra al final de la estructura articulada del robot, generalmente se trata de algún elemento de sujeción o alguna herramienta. Dentro de los elementos de sujeción encontramos pinzas de presión o enganche tanto lineal o angular, electroimanes o ventosas de vacío. Para el caso de efector final del tipo herramienta existe gran variedad de estas mismas. Ya sea para tareas como soldar, atornillar, lijar, desbastar, cortar, pintar, entre otras.

Los elementos de sujeción del tipo gripper, pertenecen a un subsistema de un robot que mantiene un número limitado de piezas geométricamente definidas durante un periodo de tiempo determinado [Wolf et al. (19)].

2.1.9. Programación de robots

El lenguaje de programación de robots requiere de los siguientes elementos.

- **Posicionamiento y orientación.** Para este apartado es importante que la posición se encuentre dentro del espacio de trabajo, prevenir que no haya colisiones con objetos o con el mismo robot.
- **Órdenes de movimiento.** Se debe considerar las velocidad, aceleración y desaceleración, estas dependen del tipo de aplicación a realizar. Para ello es necesario consultar fichas técnicas. Por ejemplo, un cortador en su hoja de datos establece una velocidad de avance ideal, así como una profundidad de corte. También pueden contener datos de velocidades de entrada y salida.
- **Sincronización de sucesos.** Esto viene más a la parte de automatización, mediante señales proporcionadas por sensores, el robot atiende algún suceso, tal como activación, espera, terminación, anulación.
- **Secuencia en las instrucciones.** La memoria del controlador debe ser capaz de almacenar datos de posiciones y acciones, debido a que sin ello se tendrían que cargar constantemente los datos al programa y esto tanto ocupa memoria como se traduce en pérdida de tiempo. Por ello a principios del programa se definen velocidades, tiempos, posición del efector final, aceleraciones, desaceleraciones, el espacio de trabajo, para después realizar secuencialmente y continuamente el demás programa sin interrupciones.

2.2. Sistemas CAD/CAM automatizados

El término CAD/CAM significa diseño asistido por computadora y manufactura asistida por computadora respectivamente. De forma general, CAD se traduce a la elaboración de diseños, dibujos, modelos ya sea en 2D o 3D, a través de una computadora. CAM, como tal, hace referencia a las acciones que lleva a cabo la maquinaria por control numérico para que se realicen los diseños CAD en piezas físicas, se especifican las instrucciones que se le darán a las maquinas, y se puede ejecutar simulaciones de las operaciones que se realicen.

Los desarrolladores de esta tecnología fueron del Instituto de Tecnología de Massachusetts (MIT) a finales de los años 40's y principios de los 50's del siglo pasado, con fines de fabricación a las industrias aeroespaciales después de la segunda guerra mundial. A mediados de los 60's se ocuparon computadoras centrales, se le llamó Control Numérico Directo (DNC). Para finales de los 60's comienza a introducirse el término CNC, donde las computadoras podían almacenar varios programas, editarlos fuera de línea. Desde los 70's los controladores numéricos se diseñan por medio de microprocesadores, hasta la actualidad que se desarrolla

con microprocesadores de 32 y 64 bits [Radhakrishnan et al. (14)].

Durante la historia del control numérico hubo ventajas en la producción, ya que las maquinarias eran capaces de producir hasta 5 veces más en comparación con la producción de manera manual. La llegada del CNC fue el auge en la manufactura, no solo por el hecho de producir más, sino por ofrecer mayor flexibilidad debido a que se pueden realizar modificaciones a los diseños de forma fácil y permite la realización de distintos lotes de diseños.

El programa de control numérico, para realizar movimientos en formas de línea recta o arco, hace uso de la interpolación para aproximar que los actuadores cumplan la trayectoria especificada. La exactitud del movimiento depende tanto de la máquina CNC, como del software CAM. Las máquinas CNC que se encuentran en el mercado son: tornos, fresadoras, centros de maquinado, máquinas de soldar, máquinas de oxicorte, rectificadoras, máquinas de corte por agua, laser, plasma, entre otros.

2.3. Sistema CNC

Es un sistema que sirve para controlar una máquina mediante una computadora, se basa en el control la posición en el espacio 3D (X, Y, Z) de un elemento físico, generalmente la herramienta de trabajo. Mediante un conjunto de ordenes se controla la posición actual con respecto a un punto de origen. Pero no sólo se puede controlar la posición, sino también la velocidad, entre otros parámetros.

2.4. Código G y M

Se le conoce como código G y M al lenguaje de programación para maquinado. Este lenguaje es de bajo nivel, secuencial y contiene la información necesaria para lograr el maquinado de una pieza a través de una máquina CNC. En un código G se especifican la funciones generales de movimiento de la máquina (movimientos, avances, ciclos, pausas, interpolación, etc.) y en un código M se especifican las funciones misceláneas requeridas para el maquinado (arranque del husillo, paro del husillo, cambio de herramienta, paro de programa, etc.).

Capítulo 3

Análisis cinemático y dinámico del robot SCORBOT ER-III con 3 DOF

3.1. Cinemática directa

La cinemática es una rama de la mecánica que estudia las propiedades de los movimientos físicos (posición, velocidad, aceleración) sin hacer consideración de las fuerzas que ocasionan esto [Craig (8)]. Para el caso de un brazo robótico, se necesita calcular y orientar el efector final relativa al eslabón base.

3.2. Parámetros Denavit Hartenberg (PDH)

Denavit y Hartenberg propusieron en 1955 un método que describe y representa la geometría espacial de un robot respecto a un sistema de referencia. Tal método, utiliza una matriz de transformación homogénea de 4×4 que describe la relación espacial entre dos elementos rígidos adyacentes. En este método se localiza tanto la posición como la orientación de un robot. Mediante 4 transformaciones dependientes de las características geométricas de cada eslabón y fijando adecuadamente los sistemas de coordenadas, es posible pasar de un eslabón a otro. Estas transformaciones consisten en traslaciones y rotaciones que se hacen de un elemento i a un elemento $i-1$ [Desai (9)] [Siciliano et al. (18)]. Las transformaciones son las siguientes:

Rotación junto del eje Z_{i-1} un ángulo θ_i

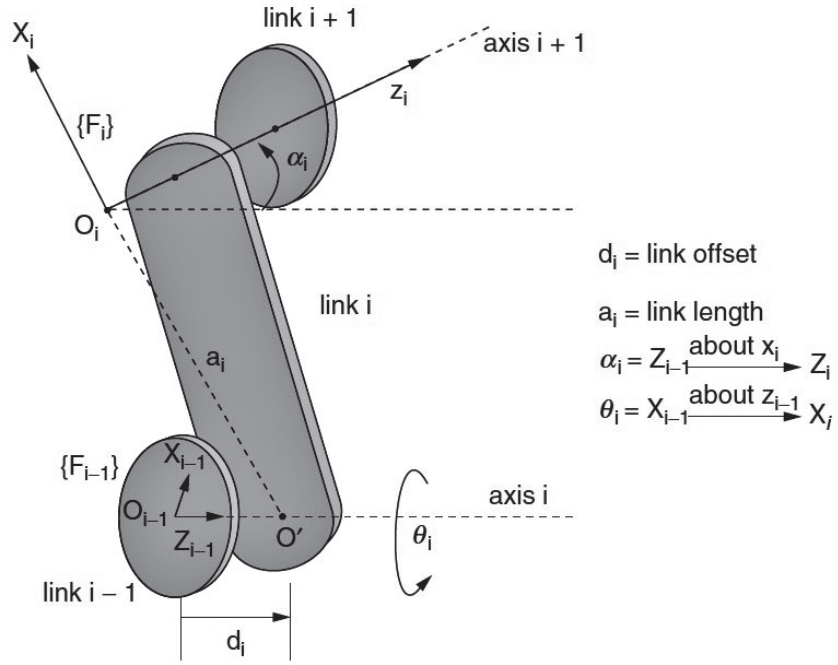


Figura 3.1: Parámetros Denavit-Hartenberg

Traslación a lo largo de Z_{i-1} una distancia d_i

Traslación a lo largo de X_i una distancia a_i

Rotación a lo largo del eje X_i un ángulo α_i

De la Figura 3.1 se observa que los marcos de cada eslabón han sido establecidos, con ello las posiciones i respecto a su posición anterior $i-1$, se especifican mediante los siguientes parámetros:

- θ_i: Es el ángulo que hay que girar en torno a Z_{i-1} para que X_{i-1} y X_i queden paralelos.
- d_i: Es la distancia medida entre X_i y X_{i-1} a lo largo de Z_{i-1}.
- a_i: Distancia entre el punto de origen O_i y el punto O_{i+1}.
- α_i: Ángulo entre los ejes Z_{i-1} y Z_i a través del eje X_i para ser positivo cuando la rotación va en sentido a las manecillas del reloj.

3.3. Algoritmo para determinar la matriz de transformación homogénea, A_n

Después de determinar los PDH, el siguiente paso es obtener la matriz de transformación homogénea $\mathbf{A}_n$ desde un eslabón al siguiente [Desai (9)]. Denotaremos la matriz $\mathbf{U}_i(\theta_i)$ como la matriz de rotación sobre el eje Z_{i-1} mientras que la matriz $\mathbf{V}_i(\alpha_i)$ como la matriz de rotación que denota la rotación sobre el eje X_i. Para realizar la transformación de coordenadas del marco de referencia F_i al marco de referencia F_{i-1} necesitamos derivar la matriz de rotación y el vector de desplazamiento. La matriz de rotación desde F_i a F_{i-1} está dada por:

$$\mathbf{R}_i^{i-1} = \mathbf{U}_i \mathbf{V}_i, \tag{3.1}$$

donde $\mathbf{U}_i(\theta_i)$ está dado por:

$$\mathbf{U}_i(\theta_i) = \begin{bmatrix} \cos(\theta_i) & -\sin(\theta_i) & 0 \\ \sin(\theta_i) & \cos(\theta_i) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

y $\mathbf{V}_i(\alpha_i)$ está dado por:

$$\mathbf{V}_i(\alpha_i) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\alpha_i) & -\sin(\alpha_i) \\ 0 & \sin(\alpha_i) & \cos(\alpha_i) \end{bmatrix}$$

la matriz de tranformación homogénea que transforma las coordenadas desde F_i a F_{i-1} está dado por:

$$\mathbf{A}_i^{i-1} = \mathbf{A}_{i'}^{i-1} \mathbf{A}_i^{i'}, \tag{3.2}$$

donde $\mathbf{A}_i^{i'}$ es el giro sobre el eje x_i mediante un ángulo α_i y una distancia a_i. Similarmente, $\mathbf{A}_{i'}^{i-1}$ es el giro sobre el eje z_i mediante un ángulo θ_i y una distancia d_i. La expresión para $\mathbf{A}_{i'}^{i-1}$ y $\mathbf{A}_i^{i'}$ está dada por:

$$\mathbf{A}_{i'}^{i-1} = \begin{bmatrix} & & 0 \\ & \mathbf{U}_{i(3\times3)} & 0 \\ & & d_i \\ 0_{(1\times3)} & & 1 \end{bmatrix}$$

$$\mathbf{A}_{i}^{i'} = \begin{bmatrix} & & a_i \\ & \mathbf{V}_{i(3\times3)} & 0 \\ & & 0 \\ 0_{(1\times3)} & & 1 \end{bmatrix}$$

consequentemente, de (3.2)

$$\mathbf{A}_{i}^{i-1} = \begin{bmatrix} & & 0 \\ & \mathbf{U}_{i(3\times3)} & 0 \\ & & d_i \\ 0_{(1\times3)} & & 1 \end{bmatrix} \begin{bmatrix} & & a_i \\ & \mathbf{V}_{i(3\times3)} & 0 \\ & & 0 \\ 0_{(1\times3)} & & 1 \end{bmatrix} = \begin{bmatrix} \mathbf{U}_i\mathbf{V}_{i(3\times3)} & \mathbf{U}_i\mathbf{s}_{i(3\times1)} \\ 0_{(1\times3)} & 1 \end{bmatrix}$$

donde el vector $\mathbf{s}_i$ se estructura como:

$$\mathbf{s}_i = \begin{bmatrix} a_i \\ 0 \\ d_i \end{bmatrix}$$

$\mathbf{U}_i\mathbf{V}_i$ es la matriz de orientación y $\mathbf{U}_i\mathbf{s}_i$ es el vector de posición. Ahora para un sistema de n DOF tenemos:

$$\mathbf{A}_n^0 = \mathbf{A}_1^0\mathbf{A}_2^1...\mathbf{A}_n^{n-1} = \begin{bmatrix} \mathbf{R}_{n(3\times3)}^0 & \mathbf{d}_{n(3\times1)}^0 \\ 0_{(1x3)} & 1 \end{bmatrix}$$

donde la matriz $\mathbf{R}_{n(3\times3)}^0$ se define como:

$$\mathbf{R}_{n(3\times3)}^0 = (\mathbf{U}_1\mathbf{V}_1)(\mathbf{U}_2\mathbf{V}_2)...(\mathbf{U}_n\mathbf{V}_n) \tag{3.3}$$

mientras que el vector $\mathbf{d}_{n(3\times1)}^0$ está dado como:

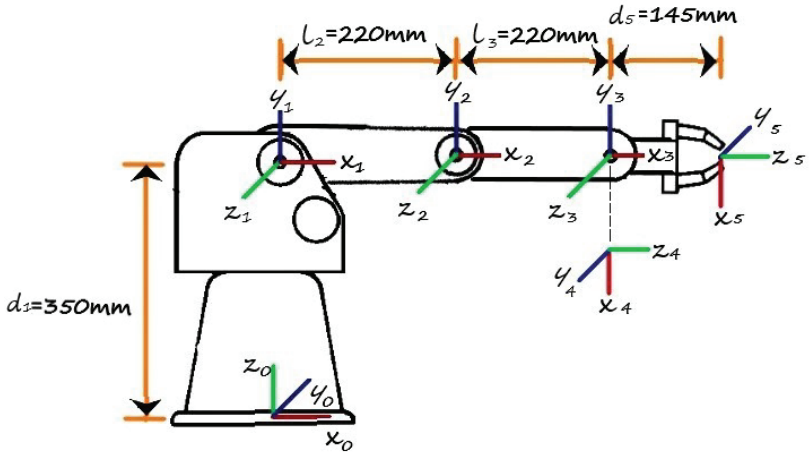


Figura 3.2: Representacion de los PDH Scorbot ER-III

$$\mathbf{d}^0_{n(3\times1)} = \mathbf{U}_1\mathbf{s}_1+\mathbf{U}_1\mathbf{V}_1\mathbf{U}_2\mathbf{s}_2+\mathbf{U}_1\mathbf{V}_1\mathbf{U}_2\mathbf{V}_2\mathbf{U}_3\mathbf{s}_3...+\mathbf{U}_1\mathbf{V}_1...\mathbf{U}_{n-1}\mathbf{V}_{n-1}\mathbf{U}_n\mathbf{s}_n \quad (3.4)$$

Para el robot SCORBOT ER-III, se obtiene los valores de la Tabla 3.1 tomando de referencia la Figura 3.2, para después obtener las matrices de transformación homogénea $\mathbf{A}_n$.

Parámetros				
	θ_i	α_i	a_i	d_i
$1_{(0-1)}$	θ_1	$-\frac{\pi}{2}$	0	d_1
$2_{(1-2)}$	θ_2	0	l_2	0
$3_{(2-3)}$	θ_3	0	l_3	0
$4_{(3-4)}$	θ_4	$\frac{\pi}{2}$	0	0
$5_{(4-5)}$	θ_5	0	0	d_5

Tabla 3.1: Parámetros obtenidos para el robot SCORBOT ER-III.

de acuerdo a los PDH que se tienen en la Tabla 3.1 se obtienen las matrices $\mathbf{A}_{i(4\times4)}^{i-1}$ como:

$$\mathbf{A}_0^1 = \begin{bmatrix} C_1 & 0 & S_1 & 0 \\ S_1 & 0 & -C_1 & 0 \\ 0 & -1 & 0 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{A}_1^2 = \begin{bmatrix} C_2 & -S_2 & 0 & l_2C_2 \\ S_2 & C_2 & 0 & l_2S_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{A}_2^3 = \begin{bmatrix} C_3 & -S_3 & 0 & l_3C_3 \\ S_3 & C_3 & 0 & l_3S_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{A}_3^4 = \begin{bmatrix} C_4 & 0 & S_4 & 0 \\ S_4 & 0 & -C_4 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{A}_4^5 = \begin{bmatrix} C_5 & -S_5 & 0 & 0 \\ S_5 & C_5 & 0 & 0 \\ 0 & 0 & 1 & d_5 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

donde:

$$C_i = \cos(\theta_i) \qquad\qquad S_i = \sin(\theta_i)$$

Al realizar el producto entre las matrices para conocer la posición y orientación desde la posición del efector final al punto de origen, se obtiene para la matriz de rotación $\mathbf{R}_{0(3\times3)}^5$, los siguientes elementos:

$$\mathbf{R}_5^0 = \begin{bmatrix} R_{(1,1)} & R_{(1,2)} & R_{(1,3)} \\ R_{(2,1)} & R_{(2,2)} & R_{(2,3)} \\ R_{(3,1)} & R_{(3,2)} & R_{(3,3)} \end{bmatrix}$$

$$\begin{aligned}
R_{(1,1)} &= C_1C_5[C_4(C_2C_3 - S_2S_3) - S_4(C_2S_3 + C_3S_2)] + S_1S_5 \\
R_{(1,2)} &= -C_1S_5[C_4(C_2C_3 - S_2S_3) - S_4(C_2S_3 + C_3S_2)] - C_5S_1 \\
R_{(1,3)} &= C_1[S_4(C_2C_3 - S_2S_3) - C_4(C_2S_3 + C_3S_2)] \\
R_{(2,1)} &= C_5S_1[C_4(C_2C_3 - S_2S_3) - S_4(C_2S_3 + C_3S_2)] + C_1S_5 \\
R_{(2,2)} &= -S_1S_5[C_4(C_2C_3 - S_2S_3) - S_4(C_2S_3 + C_3S_2)] + C_1C_5 \\
R_{(2,3)} &= S_1[S_4(C_2C_3 - S_2S_3) - C_4(C_2S_3 + C_3S_2)] \\
R_{(3,1)} &= -C_5[C_4(C_3S_2 + C_2S_3) + S_4(C_2C_3 - S_2S_3)] \\
R_{(3,2)} &= S_5[C_4(C_3S_2 + C_2S_3) + S_4(C_2C_3 - S_2S_3)] \\
R_{(3,3)} &= -S_4(C_3S_2 + C_2S_3) + C_4(C_2C_3 - S_2S_3)
\end{aligned}$$

y el vector de posición $\mathbf{d}_{0(3\times1)}^5$ resulta:

$$\mathbf{d}_5^0 = \begin{bmatrix} d_{(1,1)} \\ d_{(2,1)} \\ d_{(3,1)} \end{bmatrix}$$

$$\begin{aligned}
d_{(1,1)} &= C_1[(l_3 + S_4d_5)(C_2C_3 - S_2S_3) + l_2C_2 + C_4d_5(S_2S_3 + C_3S_2)] \\
d_{(2,1)} &= S_1[(l_3 + S_4d_5)(C_2C_3 - S_2S_3) + l_2C_2 + C_4d_5(C_2S_3 + C_3S_2)] \\
d_{(3,1)} &= -[(l_3 + S_4d_5)(C_2S_3 + C_3S_2) + C_4d_5(S_2S_3 - C_2C_3) + l_2S_2 + d_1]
\end{aligned}$$

Comentario 3.3.1 *La matriz de orientación y el vector de posición sirven para determinar las coordenadas operacionales del robot en X, Y, Z en función de las variables articulares θ_i. Además se puede hallar la cinemática inversa para el robot. En este trabajo sólo se presentan los cáculos de obtención de la matriz de rotación y el vector de posición para trabajos posteriores.*

3.4. Cinemática inversa

En la cinemática inversa se resuelven los cálculos de todos los posibles ángulos entre eslabones, para que el efector final llegue a una posición y orientación en

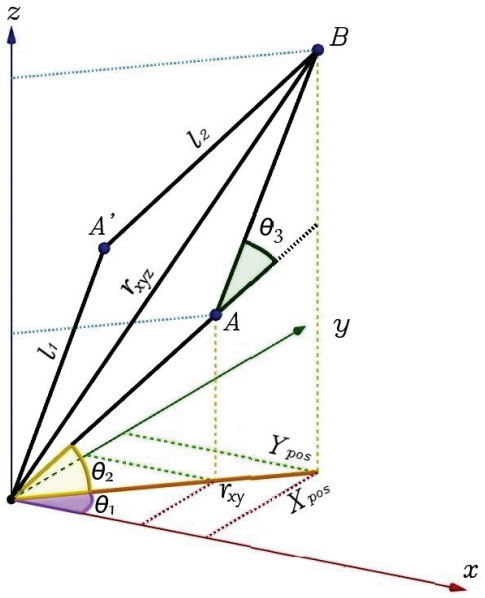


Figura 3.3: Multiples soluciones de ángulos en un robot

el espacio 3D. La resolución de estos ángulos resulta ser el tema más importante y a su vez difícil para diseñar un robot, esto por el hecho que se hace uso de ecuaciones cinemáticas no lineales y no siempre es posible encontrar solución o pueden existir múltiples soluciones [Saha (16)]. El hecho que exista una o múltiples soluciones significa que la posición y orientación del efector final está dentro del espacio de trabajo.

Para la solución de estos ángulos existen varios métodos, los más comunes son el método algebraico y el método geométrico. En cualquier caso se debe procurar simplificar el número de operaciones que se realizan, para que al microprocesador le tome el menor tiempo de procesar los datos.

En este trabajo se optó por el método de solución geométrico basado en relaciones trigonométricas. Es bien sabido que para 3 DOF existen dos posibles soluciones para llegar a una posición X, Y, Z, como se muestra en la Figura 3.3. Una solución es cuando el codo está abajo (punto A), y otra es cuando el codo esta arriba (punto A').

Para el cálculo de el ángulo θ_1:

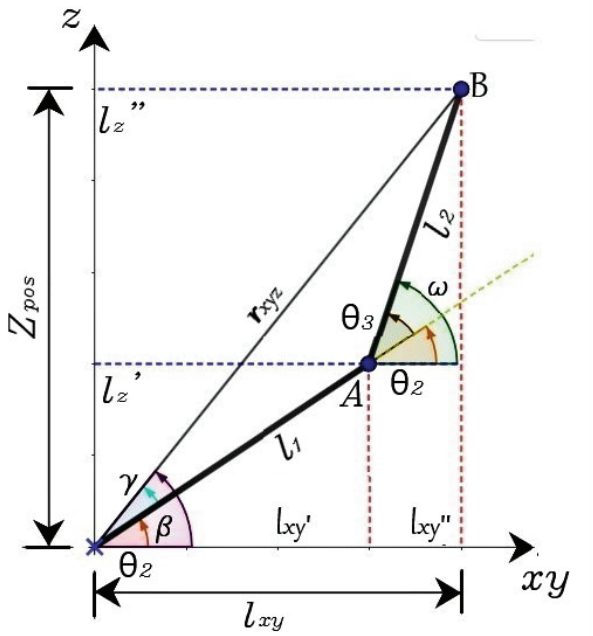


Figura 3.4: Solución cinemática inversa codo abajo

$$\theta_1 = \tan^{-1}\left(\frac{Y_{pos}}{X_{pos}}\right) \tag{3.5}$$

Para el cálculo de el ángulo θ_2, dada la Figura 3.4:

$$r_{xy} = \frac{X_{pos}}{\cos\theta_1} \tag{3.6}$$

$$\beta = \tan^{-1}\left(\frac{Z_{pos}}{r_{xy}}\right) \tag{3.7}$$

$$\gamma = \cos^{-1}\left(\frac{l_1^2 + l_2^2\sqrt{r_{xy}^2 Z_{pos}^2}}{2l_1\sqrt{r_{xy}^2 Z_{pos}^2}}\right) \tag{3.8}$$

Por lo tanto:

$$\theta_2 = \beta - \gamma \tag{3.9}$$

En el caso de la Figura 3.5, θ_2 sólo cambia de resta a suma entre β y γ.

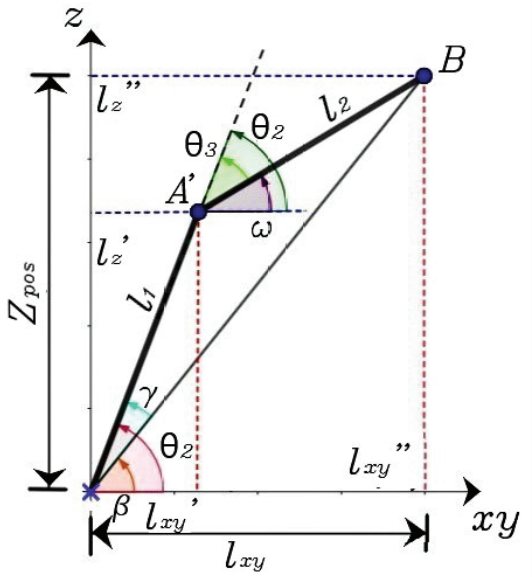


Figura 3.5: Solución cinemática inversa codo arriba

Para el cálculo de el ángulo θ_3, basado en la Figura 3.4:

$$l'_y = l_1 \sin\theta_2 \tag{3.10}$$

$$l'_x = l_1 \cos\theta_2 \tag{3.11}$$

$$l''_{xy} = l_{xy} - l'_x \tag{3.12}$$

$$l''_z = Z_{pos} - l'_y \tag{3.13}$$

$$\omega = \tan^{-1}\left(\frac{l''_{xy}}{l''_z}\right) \tag{3.14}$$

$$\theta_3 = \omega - \theta_2 \tag{3.15}$$

En el caso de codo arriba, basando en la Figura 3.5, los cálculos realizados hasta (3.14) son los mismos, solo que ahora θ_3 resulta:

$$\theta_3 = \theta_2 - \omega \tag{3.16}$$

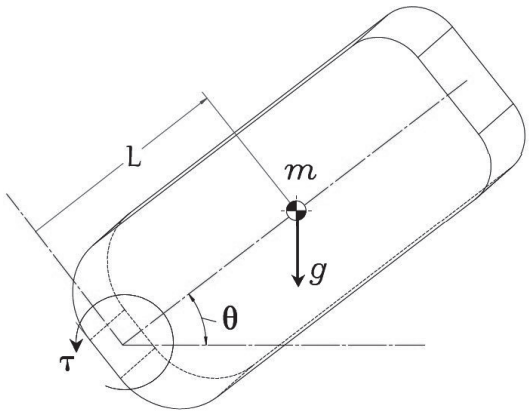


Figura 3.6: Inercia en un eslabón

3.5. Dinámica de un robot

La dinámica es una área de la mecánica que estudia las fuerzas que ocasionan los movimientos. Para el caso de un manipulador, el modelo dinámico permite explicar todos lo fenómenos físicos que se encuentran en su estructura mecánica, tales como efectos inerciales, fuerzas centrípetas y de Coriolis, par gravitacional y fricción [Reyes (15)]. Una vez conociendo el conjunto de momentos de torsión del actuador se puede lograr mantenerlo a una velocidad constante, acelerarlo o desacelerarlo [Craig (8)].

3.5.1. Modelo dinámico de un robot

El modelo dinámico de un robot es la relación entre entre el movimiento y las fuerzas involucradas en el robot. Este modelo es basado en el equlibrio de fuerzas de Newton o su equivalente para movimientos rotacionales de Euler. La Figura 3.6 muestra el caso de un simple eslabón, la suma de pares-fuerzas que actuan sobre el eslabón son los de la siguiente ecuación.

$$\tau - mgL\cos\theta = I\frac{d^2\theta}{dt^2} \tag{3.17}$$

$$\tau = mL^2\ddot{\theta} + mgL\cos\theta \tag{3.18}$$

Con:

$$\ddot{\theta} = \frac{d^2\theta}{dt^2}$$

donde cada uno de los parámetros son los siguientes:

- τ: par aplicado al eslabón.
- m: masa del eslabón.
- L: longitud del centro de giro hacia el centro de masa del eslabón.
- θ: ángulo de giro.
- I: tensor de inercia del eslabón alrededor del centro de masa.
- g: el vector 3D de aceleración debido a la gravedad.

Existen dos diferentes tipos de modelos dinámicos para un brazo robótico, el directo y el inverso:

- Dinámica directa. Expresa la evolución temporal (de acuerdo al tiempo transcurrido) de las coordenadas de la articulaciones de un robot en función de las fuerzas y/o pares involucrados para hacer que se produzca movimiento.
- Dinámica inversa. Determina las fuerzas y/o pares necesarios para lograr la evolución conjunta de las coordenadas determinadas.

Existe también otro método para obtener el modelo dinámico de un robot, basada en consideraciones energéticas. Por Euler-Lagrange se obtiene el modelo dinámico directo de un robot, donde se consideran los siguientes pasos:

- 1. Obtener la energía cinética de los eslabones:

$$K_i(\mathbf{q}, \dot{\mathbf{q}}) = \frac{1}{2} m_i v^2 + \frac{1}{2} I_i \dot{\theta}^2 \tag{3.19}$$

- 2. Obtener la energía potencial de los eslabones:

$$U_i(\mathbf{q}) = m_i g h \tag{3.20}$$

- 3. Obtener el lagrangiano del sistema. Que es la diferencia de la energía cinética total y la energía potencial total.

$$\mathcal{L}(\mathbf{q}, \dot{\mathbf{q}}) = K(\mathbf{q}, \dot{\mathbf{q}}) - U(\mathbf{q}) \tag{3.21}$$

- 4. Desarrollar las ecuaciones de Lagrange:

$$\frac{d}{dt}\left(\frac{\partial \mathcal{L}}{\partial \dot{\theta}_i}\right) - \frac{\partial \mathcal{L}}{\partial \theta_i} = \tau_i \tag{3.22}$$

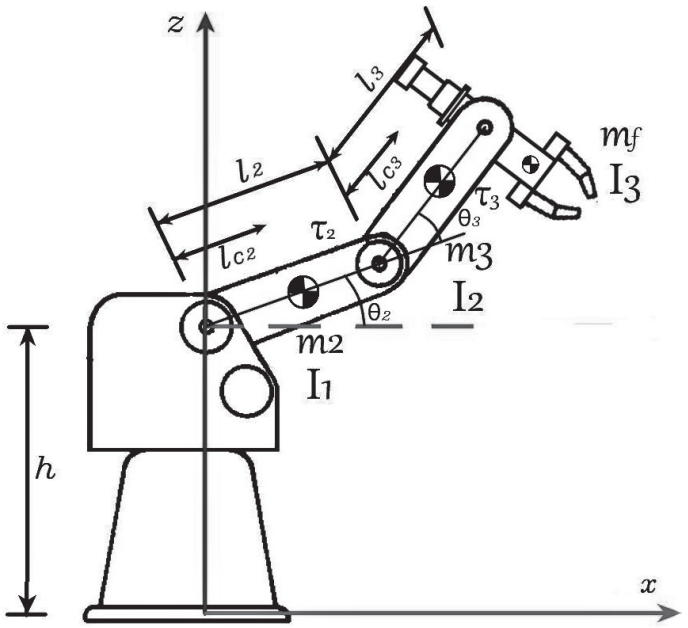


Figura 3.7: Representación dinámica para el robot SCORBOT ER-III

Para el robot SCORBOT ER-III basado en la Figura 3.7, se tiene para el eslabón 1 [Martinez et al. (13)]:

$$K_1 = \frac{1}{2} I_1 \dot{\theta_1}^2 \tag{3.23}$$

$$U_1 = m_1 g h \tag{3.24}$$

Para el eslabón 2:

$$K_2 = \frac{1}{2} m_2 \left[l_{c2}^2 \dot{\theta_2}^2 + l_{c2}^2 \cos^2 \theta_2 \dot{\theta_1}^2 \right] + \frac{1}{2} I_2 \left(\dot{\theta_1}^2 + \dot{\theta_2}^2 \right) \tag{3.25}$$

$$U_2 = m_2 g \left(h + l_{c2} \sin \theta_2 \right) \tag{3.26}$$

Para el eslabón 3:

$$\begin{aligned} K_3 &= \frac{1}{2}m_3 l_{c3}^2 \dot{\theta_2}^2 + \frac{1}{2}m_3 l_{c3}^2 \dot{\theta_3}^2 + m_3 l_{c3}^2 \dot{\theta_2}\dot{\theta_3} + m_3 l_2 l_{c3}\cos(\theta_2)\cos(\theta_2+\theta_3)\dot{\theta_1}^2 \\ &\quad + \frac{1}{2}m_3 l_2^2 \dot{\theta_2}^2 + \frac{1}{2}m_3 l_{c3}^2 \cos^2(\theta_2+\theta_3)\dot{\theta_1}^2 + \frac{1}{2}m_3 l_2^2 \cos^2(\theta_2)\dot{\theta_1}^2 \\ &\quad + m_3 l_2 l_{c3}\cos(\theta_3)\dot{\theta_2}(\dot{\theta_2}+\dot{\theta_3}) + \frac{1}{2}I_3\left[\dot{\theta_1}^2 + (\dot{\theta_2}+\dot{\theta_3})^2\right] \qquad (3.27) \\ U_3 &= m3g(h + l_2\sin\theta_2 + l_{c3}\sin(\theta_2+\theta_3)) \qquad (3.28) \end{aligned}$$

Para el efector final (gripper):

$$\begin{aligned} K_4 &= \frac{1}{2}m_f l_3^2 \dot{\theta_2}^2 + \frac{1}{2}m_f l_3^2 \dot{\theta_3}^2 + m_f l_3^2 \dot{\theta_2}^2 \dot{\theta_3}^2 + m_f l_2 l_3 \cos(\theta_2)\cos(\theta_2+\theta_3)\dot{\theta_1}^2 \\ &\quad + \frac{1}{2}m_f l_2^2 \dot{\theta_2}^2 + \frac{1}{2}m_f l_3^2 \cos^2(\theta_2+\theta_3)\dot{\theta_1}^2 + \frac{1}{2}m_f l_2^2 \cos^2(\theta_2)\dot{\theta_1}^2 \\ &\quad + m_f l_2 l_3 \cos(\theta_3)\dot{\theta_2}(\dot{\theta_2}+\dot{\theta_3}) \qquad (3.29) \\ U_4 &= m_f g\,(h + l_2\sin\theta_2 + l_2\sin(\theta_2+\theta_3)) \qquad (3.30) \end{aligned}$$

El Lagrangiano del sistema es:

$$\begin{aligned} \mathcal{L} &= \frac{1}{2}I_1\dot{\theta_1}^2 + \frac{1}{2}m_2\left[l_{c2}^2\dot{\theta_2}^2 + l_{c2}^2\cos^2\theta_2\dot{\theta_1}^2\right] + \frac{1}{2}I_2\left(\dot{\theta_1}^2 + \dot{\theta_2}^2\right) + \frac{1}{2}m_3 l_{c3}^2\dot{\theta_2}^2 \\ &\quad + \frac{1}{2}m_3 l_{c3}^2\dot{\theta_3}^2 + m_3 l_{c3}^2\dot{\theta_2}\dot{\theta_3} + m_3 l_2 l_{c3}\cos(\theta_2)\cos(\theta_2+\theta_3)\dot{\theta_1}^2 + \frac{1}{2}m_3 l_2^2\dot{\theta_2}^2 \\ &\quad + \frac{1}{2}m_3 l_{c3}^2\cos^2(\theta_2+\theta_3)\dot{\theta_1}^2 + \frac{1}{2}m_3 l_2^2\cos^2(\theta_2)\dot{\theta_1}^2 m_3 l_2 l_{c3}\cos(\theta_3)\dot{\theta_2}(\dot{\theta_2}+\dot{\theta_3}) \\ &\quad + \frac{1}{2}I_3\left[\dot{\theta_1}^2 + (\dot{\theta_2}+\dot{\theta_3})^2\right] + \frac{1}{2}m_f l_3^2\dot{\theta_2}^2 + \frac{1}{2}m_f l_3^2\dot{\theta_3}^2 + m_f l_3^2\dot{\theta_2}^2\dot{\theta_3}^2 + \frac{1}{2}m_f l_2^2\dot{\theta_2}^2 \\ &\quad + m_f l_2 l_3\cos(\theta_2)\cos(\theta_2+\theta_3)\dot{\theta_1}^2 + \frac{1}{2}m_f l_3^2\cos^2(\theta_2+\theta_3)\dot{\theta_1}^2 + \frac{1}{2}m_f l_2^2\cos^2(\theta_2)\dot{\theta_1}^2 \\ &\quad + m_f l_2 l_3\cos(\theta_3)\dot{\theta_2}(\dot{\theta_2}+\dot{\theta_3}) - m_1 g h - m_3 g(h + l_2\sin\theta_2 + l_{c3}\sin(\theta_2+\theta_3)) \\ &\quad - m_2 g\,(h + l_{c2}\sin\theta_2) - m_f g\,(h + l_2\sin\theta_2 + l_2\sin(\theta_2+\theta_3)) \qquad (3.31) \end{aligned}$$

Desarrollando la ecuación para cada articulación, se obtiene:

$$\begin{aligned}
\tau_1 &= I_1\ddot{\theta_1} + I_2\ddot{\theta_1} + I_3\ddot{\theta_1} + m_2 l_{c2}^2 \cos^2(\theta_2)\ddot{\theta_1} - 2m_2 l_{c2}^2 \cos\theta_2 \sin\theta_2 \dot{\theta_1}\dot{\theta_2} \\
&\quad - 2m_3 l_2 l_{c3} \sin\theta_2 \cos(\theta_2+\theta_3)\dot{\theta_1}\dot{\theta_2} - 2m_3 l_2 l_{c3} \cos\theta_2 \sin(\theta_2+\theta_3)\dot{\theta_1}(\dot{\theta_2}+\dot{\theta_3}) \\
&\quad + 2m_3 l_2 l_{c3} \cos\theta_2 \cos(\theta_2+\theta_3)\ddot{\theta_1} - 2m_3 l_3 l_{c3}^2 \sin(\theta_2+\theta_3)\cos(\theta_2+\theta_3)\dot{\theta_1}(\dot{\theta_2}+\dot{\theta_3}) \\
&\quad + m_3 l_{c3}^2 \cos^2(\theta_2+\theta_3)\ddot{\theta_1} - 2m_3 l_2^2 \sin\theta_2 \cos\theta_2 \dot{\theta_1}\dot{\theta_2} + m_3 l_2^2 \cos^2(\theta_2)\ddot{\theta_1} \\
&\quad - 2m_f l_2 l_3 \sin\theta_2 \cos(\theta_2+\theta_3)\dot{\theta_1}\dot{\theta_2} - 2m_f l_2 l_3 \cos\theta_2 \sin(\theta_2+\theta_3)\dot{\theta_1}(\dot{\theta_2}+\dot{\theta_3}) \\
&\quad + 2m_f l_2 l_3 \cos\theta_2 \cos(\theta_2+\theta_3)\ddot{\theta_1} - 2m_f l_3^2 \sin(\theta_2+\theta_3)\cos(\theta_2+\theta_3)\dot{\theta_1}(\dot{\theta_2}+\dot{\theta_3}) \\
&\quad + m_f l_3^2 \cos^2(\theta_2+\theta_3)\ddot{\theta_1} - 2m_f l_2^2 \sin\theta_2 \cos\theta_2 \dot{\theta_1}\dot{\theta_2} + m_f l_2^2 \cos^2(\theta_2)\ddot{\theta_1} \qquad (3.32)
\end{aligned}$$

$$\begin{aligned}
\tau_2 &= I_2\ddot{\theta_2} + m_2 l_{c2}^2 \ddot{\theta_2} + m_3 l_{c3}^2 \ddot{\theta_2} + m_3 l_{c3}^2 \ddot{\theta_3} + m_3 l_2^2 \ddot{\theta_2} - 2m_3 l_2 l_{c3} \sin(\theta_3)\dot{\theta_2}\dot{\theta_3} \\
&\quad + 2m_3 l_2 l_{c3} \cos(\theta_3)\ddot{\theta_2} - m_3 l_2 l_{c3} \sin(\theta_3)\dot{\theta_3}^2 + m_3 l_2 l_{c3} \cos(\theta_3)\ddot{\theta_3} \\
&\quad + \ + m_f l_3^2 \ddot{\theta_2} + I_3(\ddot{\theta_2} + \ddot{\theta_3}) - 2m_f l_2 l_3 \sin(\theta_3)\dot{\theta_2}\dot{\theta_3} + 2m_f l_2 l_3 \cos(\theta_3)\ddot{\theta_2} \\
&\quad + m_f l_3^2 \ddot{\theta_2} + m_f l_2^2 \ddot{\theta_2} + m_f l_2^2 \cos(\theta_2)\sin(\theta_2)\ddot{(\theta_1)}^2 + m_2 g l_{c2} \cos\theta_2 + m_3 g l_2 \cos\theta_2 \\
&\quad + m_3 g l_{c3} \cos(\theta_2+\theta_3) - m_f l_2 l_3 \sin(\theta_3)\dot{\theta_3}^2 + m_2 l_{c2}^2 \cos(\theta_2)\sin(\theta_2)\ddot{(\theta_1)}^2 \\
&\quad + m_3 l_2 l_{c3} \sin(\theta_2)\cos(\theta_2+\theta_3)\dot{\theta_1}^2 + m_3 l_2 l_{c3} \cos(\theta_2)\sin(\theta_2+\theta_3)\dot{\theta_1}^2 \\
&\quad + m_f l_2 l_3 \cos(\theta_3)\ddot{\theta_3} + 2m_3 l_{c3}^2 \cos(\theta_2+\theta_3)\sin(\theta_2+\theta_3)\dot{\theta_1}^2 \\
&\quad + m_3 l_2^2 \cos(\theta_2)\sin(\theta_2)\dot{\theta_1}^2 + m_f l_2 l_3 \sin(\theta_2)\cos(\theta_2+\theta_3)\dot{\theta_1}^2 \\
&\quad + m_f l_2 l_3 \cos(\theta_2)\sin(\theta_2+\theta_3)\dot{\theta_1}^2 + m_f l_3^2 \cos(\theta_2+\theta_3)\sin(\theta_2+\theta_3)\dot{\theta_1}^2 \\
&\quad + m_f g l_2 \cos\theta_2 + m_f g l_3 \cos(\theta_2+\theta_3) \qquad (3.33)
\end{aligned}$$

$$\begin{aligned}
\tau_3 &= m_3 l_{c3}^2 \ddot{\theta_3} + m_3 l_{c3}^2 \ddot{\theta_2} + m_3 l_2 l_{c3} \cos\theta_3 \ddot{\theta_2} + m_3 l_2 l_{c3} \cos(\theta_2)\sin(\theta_2+\theta_3)\dot{\theta_1}^2 \\
&\quad - m_f l_2 l_3 \sin\theta_3 \dot{\theta_2}\dot{\theta_3} + m_f l_3^2 \ddot{(\theta_3)} + m_3 l_{c3}^2 \cos(\theta_2+\theta_3)\sin(\theta_2+\theta_3)\dot{\theta_1}^2 \\
&\quad + m_f l_2 l_3 \cos(\theta_3)\ddot{\theta_2} + m_f l_3^2 \ddot{\theta_2} + m_3 l_2 l_{c3} \sin(\theta_3)\dot{\theta 2}(\dot{\theta_2}+\dot{\theta_3}) \\
&\quad + m_f l_2 l_3 \cos(\theta_2)\sin(\theta_2+\theta_3)\dot{\theta_1}^2 + m_3 g l_{c3} \cos(\theta_2+\theta_3) \\
&\quad + m_f l_3^2 \cos(\theta_2+\theta_3)\sin(\theta_2+\theta_3)\dot{\theta_1}^2 + m_f g l_3 \cos(\theta_2+\theta_3) \\
&\quad + m_f l_2 l_3 \sin(\theta_3)\dot{\theta 2}(\dot{\theta_2}+\dot{\theta_3}) \qquad (3.34)
\end{aligned}$$

Cualquier, que sea el método para obtener el modelo dinámico de un brazo robótico de n DOF, siempre tendrá la siguiente forma de una ecuación diferencial no-lineal [Reyes (15)]:

$$\tau = \mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) \tag{3.35}$$

Donde:

- $\mathbf{M}(\mathbf{q}) \in \mathbf{R}^{nxn}$ es la matriz de inercia, que significa el cambio de estado de movimiento de un robot; es simétrica y puede ser definida positiva.
- $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) \in \mathbf{R}^{nxn}$ es la matriz de fuerzas centípetras y de Coriolis. Esta última representa una desviación del movimiento de traslación debido a su componente de rotación.

$$\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} = \dot{\mathbf{M}}(\mathbf{q})\dot{\mathbf{q}} - \frac{\partial}{\partial \mathbf{q}}\left[\frac{1}{2}\dot{\mathbf{q}}^T\mathbf{M}(\mathbf{q})\dot{\mathbf{q}}\right] \tag{3.36}$$

- $\mathbf{g}(\mathbf{q}) \in \mathbf{R}^n$ Es el vector de pares gravitacionales, se obtiene con el gradiente de energía potencial:

$$\mathbf{g}(\mathbf{q}) = \frac{\partial U(\mathbf{q})}{\partial \mathbf{q}} \tag{3.37}$$

- $\mathbf{q} \in \mathbf{R}^n$ representa la posición articular del eslabón.
- $\tau \in \mathbf{R}^n$ representa el vector de pares aplicados a los eslabones:

$$\tau = [\tau_1, \tau_2, ..., \tau_n]^T \tag{3.38}$$

- n es el número DOF.

Definiendo las variables:

$$q_i = \theta_i \qquad \dot{q}_i = \dot{\theta}_i \qquad \ddot{q}_i = \ddot{\theta}_i$$

Y las matrices:

$$\mathbf{M}(\mathbf{q}) = \begin{bmatrix} M_{11}(q) & M_{12}(q) & M_{13}(q) \\ M_{21}(q) & M_{22}(q) & M_{23}(q) \\ M_{31}(q) & M_{32}(q) & M_{33}(q) \end{bmatrix}$$

$$\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) = \begin{bmatrix} C_{11}(q, \dot{q}) & C_{12}(q, \dot{q}) & C_{13}(q, \dot{q}) \\ C_{21}(q, \dot{q}) & C_{22}(q, \dot{q}) & C_{23}(q, \dot{q}) \\ C_{31}(q, \dot{q}) & C_{32}(q, \dot{q}) & C_{33}(q, \dot{q}) \end{bmatrix}$$

$$\mathbf{g}(\mathbf{q}) = \begin{bmatrix} g_1(q) & g_2(q) & g_3(q) \end{bmatrix}^T$$

El modelo dinámico en forma matricial es de la siguiente manera:

$$
\begin{aligned}
M_{11}(q) &= I_1 + I_2 + I_3 + m_2 l_{c2}^2 \cos^2(q_2) + 2m_3 l_2 l_{c3} \cos q_2 \cos(q_2+q_3) \\
&\quad + m_3 l_{c3}^2 \cos^2(q_2+q_3) + m_3 l_2^2 \cos^2(q_2) + m_f l_2^2 \cos^2(q_2) \\
&\quad + 2m_f l_2 l_3 \cos q_2 \cos(q_2+q_3) + m_f l_3^2 \cos^2(q_2+q_3) \\
M_{12}(q) &= 0 \\
M_{13}(q) &= 0 \\
M_{21}(q) &= 0 \\
M_{22}(q) &= I_2 + I_3 + m_2 l_{c2}^2 + m_3 l_{c3}^2 + m_3 l_2^2 + 2m_3 l_2 l_{c3} \cos(q_3) + m_f l_2^2 \\
&\quad + m_f l_3^2 + 2m_f l_2 l_3 \cos q_3 \\
M_{23}(q) &= I_3 + m_3 l_{c3}^2 + m_f l_3^2 + m_3 l_2 l_{c3} \cos q_3 + m_f l_2 l_3 \cos q_3 \\
M_{31}(q) &= 0 \\
M_{32}(q) &= I_3 + m_3 l_{c3}^2 + m_f l_3^2 + m_3 l_2 l_{c3} \cos q_3 + m_f l_2 l_3 \cos q_3 \\
M_{33}(q) &= I_3 + m_3 l_{c3}^2 + m_f l_3^2 \\
C_{11}(q,\dot{q}) &= -m_3 l_3 l_{c3}^2 \sin(q_2+q_3)\cos(q_2+q_3)(\dot{q}_2+\dot{q}_3) - m_3 l_2 l_{c3} \sin q_2 \cos(q_2+q_3)\dot{q}_2 \\
&\quad - m_3 l_2 l_{c3} \cos q_2 \sin(q_2+q_3)(\dot{q}_2+\dot{q}_3) - m_2 l_{c2}^2 \cos q_2 \sin q_2 \dot{q}_2 \\
&\quad - m_f l_2 l_3 \sin q_2 \cos(q_2+q_3)\dot{q}_2 - m_f l_2 l_3 \cos q_2 \sin(q_2+q_3)(\dot{q}_2+\dot{q}_3) \\
&\quad - m_f l_3^2 \sin(q_2+q_3)\cos(q_2+q_3)(\dot{q}_2+\dot{q}_3) - m_f l_2^2 \sin q_2 \cos q_2 \dot{q}_2 \\
&\quad - m_3 l_2^2 \sin q_2 \cos q_2 \dot{\theta}_2 \\
C_{12}(q,\dot{q}) &= -m_2 l_{c2}^2 \cos q_2 \sin q_2 \dot{q}_1 - m_3 l_2 l_{c3} \sin q_2 \cos(q_2+q_3)\dot{q}_1 \\
&\quad - m_3 l_2 l_{c3} \cos q_2 \sin(q_2+q_3)\dot{q}_1 - m_3 l_3 l_{c3}^2 \sin(q_2+q_3)\cos(q_2+q_3)\dot{q}_1 \\
&\quad - m_f l_3^2 \sin(q_2+q_3)\cos(q_2+q_3)\dot{q}_1 - m_3 l_2^2 \sin q_2 \cos q_2 \dot{q}_1 \\
&\quad - m_f l_2 l_3 \cos q_2 \sin(q_2+q_3)\dot{q}_1 - m_f l_2^2 \sin q_2 \cos q_2 \dot{q}_1 \\
&\quad - m_f l_2 l_3 \sin q_2 \cos(q_2+q_3)\dot{q}_1 \\
C_{13}(q,\dot{q}) &= -m_3 l_2 l_{c3} \cos q_2 \sin(q_2+q_3)\dot{q}_1 - m_3 l_3 l_{c3}^2 \sin(q_2+q_3)\cos(q_2+q_3)\dot{q}_1 \\
&\quad - m_f l_2 l_3 \cos q_2 \sin(q_2+q_3)\dot{q}_1 - m_f l_3^2 \sin(q_2+q_3)\cos(q_2+q_3)\dot{q}_1 \\
C_{21}(q,\dot{q}) &= -m_3 l_2 l_{c3} \sin q_2 \cos(q_2+q_3)\dot{q}_1 - m_3 l_2 l_{c3} \cos q_2 \sin(q_2+q_3)\dot{q}_1 \\
&\quad - m_3 l_{c3}^2 \sin(q_2+q_3)\cos(q_2+q_3)\dot{q}_1 - m_f l_2 l_3 \sin q_2 \cos(q_2+q_3)\dot{q}_1 \\
&\quad - m_2 l_{c2}^2 \cos q_2 \sin q_2 \dot{q}_1 - m_3 l_2^2 \cos q_2 \sin q_2 \dot{q}_1 - m_f l_2^2 \cos q_2 \sin q_2 \dot{q}_1 \\
&\quad - m_f l_2 l_3 \cos q_2 \sin(q_2+q_3)\dot{q}_1 - m_f l_3^2 \sin(q_2+q_3)\cos(q_2+q_3)\dot{q}_1 \\
C_{22}(q,\dot{q}) &= -m_3 l_2 l_{c3} \sin q_3 \dot{q}_3 - m_f l_2 l_3 \sin q_3 \dot{q}_3 \\
C_{23}(q,\dot{q}) &= -m_3 l_2 l_{c3} \sin q_3 \dot{q}_2 - m_3 l_2 l_{c3} \sin q_3 \dot{q}_3 - m_f l_2 l_3 \sin q_3 \dot{q}_2 - m_f l_2 l_3 \sin q_3 \dot{q}_3 \\
C_{31}(q,\dot{q}) &= m_3 l_2 l_{c3} \cos q_2 \sin(q_2+q_3)\dot{q}_1 + m_3 l_{c3}^2 \sin(q_2+q_3)\cos(q_2+q_3)\dot{q}_1 \\
&\quad + m_f l_2 l_3 \cos q_2 \sin(q_2+q_3)\dot{q}_1 + m_f l_3^2 \sin(q_2+q_3)\cos(q_2+q_3)\dot{q}_1 \\
C_{32}(q,\dot{q}) &= m_3 l_2 l_{c3} \sin q_3 (\dot{q}_2+\dot{q}_3) + m_f l_2 l_3 \sin q_3 (\dot{q}_2+\dot{q}_3) \\
C_{33}(q,\dot{q}) &= -m_3 l_2 l_{c3} \sin q_3 \dot{q}_2 - m_f l_2 l_3 \sin q_3 \dot{q}_2
\end{aligned}
$$

$$
\begin{aligned}
g_1(q) &= 0 \\
g_2(q) &= m_2 g l_{c2} \cos q_2 + m_3 g l_2 \cos q_2 + m_3 g l_{c3} \cos(q_2 + q_3) + m_f g l_2 \cos q_2 \\
&\quad + m_f g l_3 \cos(q_2 + q_3) \\
g_3(q) &= m_3 g l_{c3} \cos(q_2 + q_3) + m_f g l_3 \cos(q_2 + q_3)
\end{aligned}
$$

De [Martinez et al. (13)] se obtienen los valores de las constantes de la Figura 3.7 para la Tabla 3.2.

Parámetro	Valor	Unidad
Masa eslabón 1	7.1402	kg
Masa eslabón 2	2.2483	kg
Masa eslabón 3	1.957	kg
Masa efector final	1.0	kg
Longitud eslabón 1	0.320	m
Longitud eslabón 2	0.220	m
Longitud eslabón 3	0.220	m
Dist. centro de masa 1	0.0	m
Dist. centro de masa 2	0.052	m
Dist. centro de masa 3	0.1376	m
Momento de inercia eslabón 1	0.04624	kgm^2
Momento de inercia eslabón 2	0.02595	kgm^2
Momento de inercia eslabón 3	0.03616	kgm^2

Tabla 3.2: Constantes dinámicas para el robot SCORBOT ER-III.

Comentario 3.5.1 *En este trabajo se desarrolla por completo el análisis dinámico del robot SCORBOT ER-III, sin embargo, en el siguiente capítulo sólo se hace uso del vector de gravedad* $\mathbf{g}(\mathbf{q})$*. Para el robot DENSO VP-6442G no se hace cálculo alguno por que ya cuenta con su sistema de control.*

3.6. Sistema de control para un brazo robótico

Con el diseño de una ley de control de posición para un brazo robótico se cálcula el par necesario para lograr que el robot se mueva sin que haya retardos por cuestiones físicas. Existen diferentes leyes de control para un robot, las cuales estan diferenciadas en complejidad y efectividad; la elección de alguna ley de control depende principalmente de la tarea a realizar.

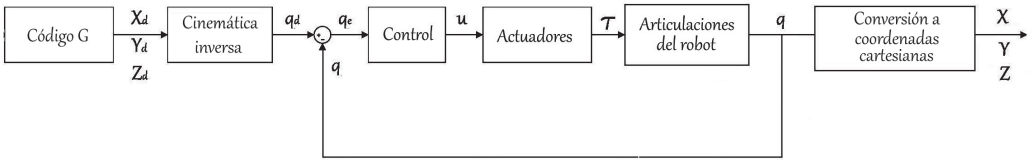


Figura 3.8: Sistema de control general para un brazo robótico

La Figura 3.8 muestra de manera general como es un sistema de control para un brazo robótico, donde la entrada a la cinemática inversa son posiciones cartesianas deseadas X_d, Y_d, Z_d y de ahí se obtienen posiciones articulares deseadas. La salida son posiciones cartesianas X, Y, Z. Sin embargo, en el sistema se controlan las posiciones articulares q. Realizada la ley de control, se obtienen los pares necesarios para los actuadores, sin embargo, a las articulaciones no se le proporciona par, sino una diferencia de potencial. Basado en el modelo de un motor de corriente directa de imán permanente se obtiene la siguiente relación entre par-voltaje [Kelly (11)].

$$u = \frac{R_a}{K_a} r\tau \tag{3.39}$$

donde:

- τ. Es el par aplicado al eslabón (Nm).
- R_a. Resistencia de armadura (Ω).
- K_a. Constante motor-par (Nm/A).
- r. Relación de reducción de los engranes $(r \gg 1)$.

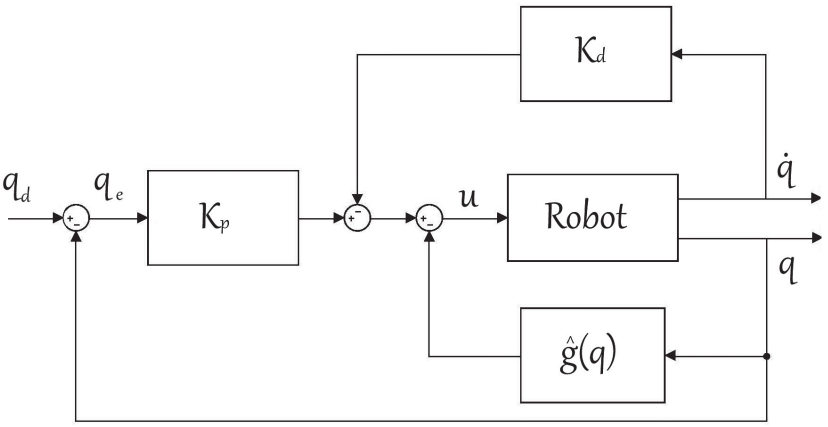


Figura 3.9: Control PD con compensación de gravedad

3.6.1. Control de posición PD con compensación de gravedad

Un robot en condiciones cuasiestaticas, es decir, se mueve lentamente o esta detenido, se puede asumir que las velocidades $\dot{q}$ y aceleraciones $\ddot{q}$ angulares son 0 [Bajd et al. (5)]. El modelo dinámico (3.35) del robot se simplifica a la siguiente ecuación:

$$\tau \approx \mathbf{g}(\mathbf{q}) \tag{3.40}$$

Esta ecuación claramente significa que los actuadores deben compensar el efecto de la gravedad.

La ley de control PD con compensación de gravedad se representa en la Figura 3.9, se puede escribir como:

$$\tau = \mathbf{K}_p(\mathbf{q}_d - \mathbf{q}) - \mathbf{K}_d\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) \tag{3.41}$$

donde $\mathbf{K}_p \in \mathbf{R}^{3x3}$, $\mathbf{K}_d \in \mathbf{R}^{3x3}$ son matrices de ganancias y $\mathbf{g}(\mathbf{q}) \in \mathbf{R}^{3x1}$ es el vector de fuerzas gravitacionales.

Esta ley de control cumple con el objetivo de control de posición conociendo unicamente del vector de gravedad, el cual puede obtenerse de la ecuación (3.37). La ley de control (3.41) requiere informacón de la posición deseada, la medición de posición y la medición de velocidad.

La ecuación que describe el comportamiento en lazo cerrado se obtiene al combinar las ecuaciones (3.41) y (3.35) [Kelly (11)].

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) = \mathbf{K}_p(\mathbf{q}_d - \mathbf{q}) - \mathbf{K}_d\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) \tag{3.42}$$

Teniendo que:

$$\widetilde{\mathbf{q}} = \mathbf{q}_d - \mathbf{q} \tag{3.43a}$$
$$\dot{\widetilde{\mathbf{q}}} = \dot{\mathbf{q}}_d - \dot{\mathbf{q}} \tag{3.43b}$$
$$\ddot{\widetilde{\mathbf{q}}} = \ddot{\mathbf{q}}_d - \ddot{\mathbf{q}} \tag{3.43c}$$

despejando $\ddot{\mathbf{q}}$ de (3.42) y sustituyendo en (3.43c), resulta:

$$\ddot{\widetilde{\mathbf{q}}} = \ddot{\mathbf{q}}_d - \mathbf{M}^{-1}\left[\mathbf{K}_p(\mathbf{q}_d - \mathbf{q}) - \mathbf{K}_d\dot{\mathbf{q}} - \mathbf{C}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}}\right] \tag{3.44}$$

y su representación en el espacio de estados de la forma $\dot{\mathbf{x}}(t) = \mathbf{A}(t)\mathbf{x}(t) + \mathbf{B}(t)\mathbf{u}(t)$, es:

$$\begin{bmatrix} \dot{\widetilde{\mathbf{q}}} \\ \ddot{\widetilde{\mathbf{q}}} \end{bmatrix} = \begin{bmatrix} \dot{\widetilde{\mathbf{q}}}_d \\ \ddot{\mathbf{q}}_d - \mathbf{M}^{-1}\left[\mathbf{K}_p(\mathbf{q}_d - \mathbf{q}) - \mathbf{K}_d\dot{\mathbf{q}} - \mathbf{C}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}}\right] \end{bmatrix}$$

Comentario 3.6.1 *El expresar la ley de control en el espacio de estados de la forma* $\dot{\mathbf{x}}(t) = \mathbf{A}(t)\mathbf{x}(t) + \mathbf{B}(t)\mathbf{u}(t)$ *sirve para realizar el análisis de estabilidad de Lyapunov en el robot, pero como el objetivo principal de este trabajo no es el sistema de control, no desarrolla la estabilidad en el robot SCORBOT ER-III.*

Capítulo 4

Análisis de Resultados

4.1. Desarrollo

A continuación se presenta el procedimiento para llevar a cabo la conversión de ambos robot a sistema CNC.

4.1.1. Implementación de la ley de control y conversión del robot SCORBOT ER-III a máquina CNC

El robot que se utilizó fue uno de la marca SCORBOT modelo ER-III de la empresa CIA Robotics que sólo contaba con su estructura mecánica, motores, transmisiones, encoders y cables de alimentación y señales. Primeramente se revisó el funcionamiento de los motores (que girarán) y que los encoders entregarán los pulsos digitales, esto se observó mediante un osciloscopio. En ocasiones la lectura de pulsos de un canal de los encoders comenzaba arriba de un voltaje de 0 Volts, es decir, tenia un offset, por lo que se prefirio hacer pruebas en un protoboard con un Circuito Integrado (IC, por sus siglas en inglés) Schmitt trigger modelo 74ls14n, con el fin de asegurar que en ambos canales las lecturas fueran 0 y 5 volts (0 y 1 digital).

Posteriormente se realizó una PCB en el software Eagle como se observa en la Figura C.8 el diseño esquemático y la Figura C.9 el diseño de la PCB incluyendo el IC Schmitt trigger y las resistencias de pull up en las entradas. En seguida se realizó un programa en el IDE de Arduino con el propósito obtener la cantidad

de pulsos en cada articulación y después relacionarlos con los ángulos de giro de cada articulación. Los resultados fueron los siguientes:

$$\theta_1 = 14,133.0 \propto \angle 330.0$$
$$\theta_2 = 5,684.0 \propto \angle 165.0$$
$$\theta_3 = 10,334.0 \propto \angle 300.0$$

Disponiendo de los pulsos proporcionados por cada encoder, se realizó un programa en el IDE de Arduino para que cada motor tuviera un sistema de control PID de un grado de libertad en lazo cerrado. Este programa solo fue para realizar un control en lazo cerrado de posición angular, el cual se ocupa posteriormente para el desarrollo de control de la primer articulación, las otras 2 articulaciones se utiliza una ley de control PD con compensación de gravedad.

Comentario 4.1.1 *Existen corrientes parásitas en los cables de comunicación entre el Arduino Mega 2560 y la computadora, lo que afecta la pérdida de información. La solución es conseguir cables que cuenten con núcleo de ferrita.*

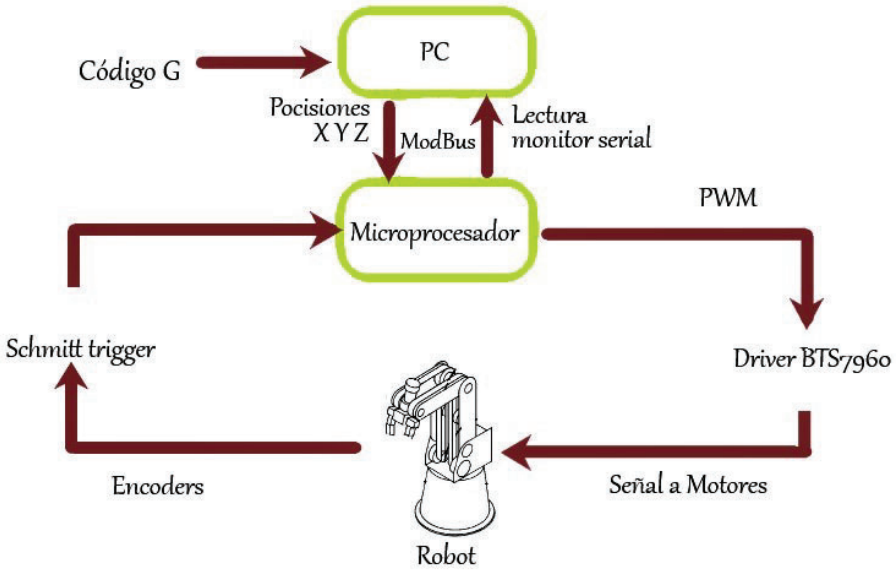


Figura 4.1: Diseño del proyecto

Teniendo que, la idea principal es a partir de un robot articulado implementar un sistema CNC (control de posición X, Y, Z de la herramienta) en tiempo real

como se observa en la Figura 4.1, se buscó un software capaz de entregar posiciones X, Y y Z en tiempo real. El software Mach3 tiene la posibilidad de transmitir estos datos mediante el protocolo Modbus. Mediante un archivo de programación en Mach3 proporcionado por la empresa CIA Robotics Figura 4.2 y modificando algunos parámetros, se procedió a realizar un programa en el IDE de Arduino para que a través del módulo FT232RL se transmitiera la información bajo el protocolo Modbus y se pudiera observar en el monitor serial la información proporcionada por Mach3.

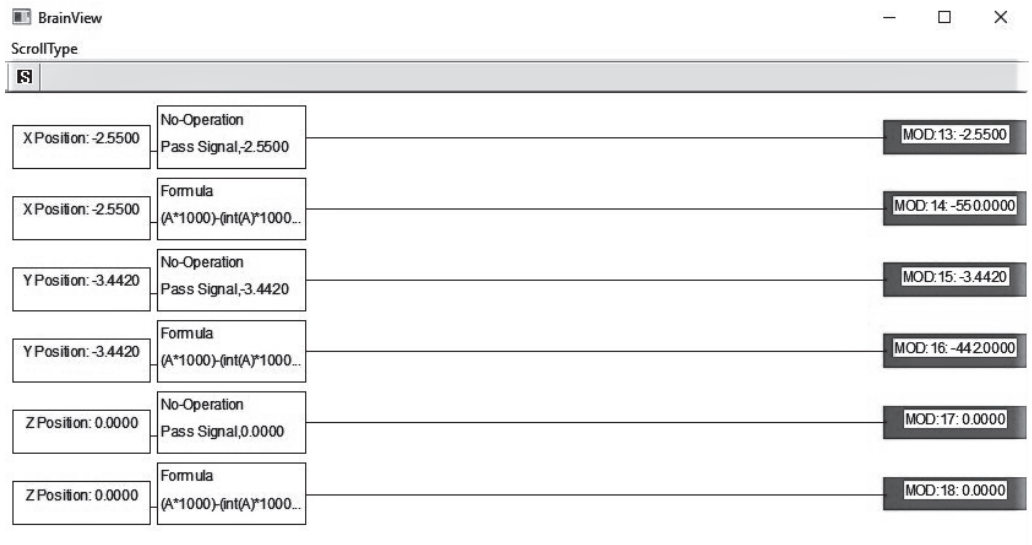


Figura 4.2: Archivo para transmitir posiciones X, Y, Z de Mach3 al IDE de Arduino

El software Mach3 ocupa sólo algún archivo en lenguaje G como entrada, este se puede generar por medio del software SheetCAM TNG Develpment, el cual a través de un dibujo en 2D logra realizar trayectorias para que se lleve a cabo el tipo de tarea. En SheetCAM se modifican los siguientes parámetros:

- Velocidad de avance
- Velocidad de entrada/salida
- Punto de entrada
- Tipo de herramienta de corte

- Distancia de penetración
- Compensación de corte, por dentro, por fuera o sobre el dibujo

Las configuraciones en Mach3 para lograr realizar la comunicación Modbus, son principalmente habilitar la comunicación en la pestaña "pines y puertos", en la pestaña "setup serial Modbus control"modificar el número de puerto y el baud rate como se observa en la Figura 4.3, los demás parámetros quedan tal cual. Para saber si el funcionamiento es correcto, en la pestaña de "test", aparece una ventana como en la Figura 4.4 y se modifica "slave addr", "num regs", después se abre el puerto y se selecciona "read", "write", si en ambos casos aparece "no error"significa que el proceso fue exitoso.

Teniendo ya la información en el microprocesador del Arduino Mega 2560, ahora se realizó un programa tanto en el IDE de Arduino, como en Matlab para calcular la cinemática inversa. Mencionar que los cálculos obtenidos son en radianes, mediante la fórmula (4.1) se hizo el cambio a grados.

$$\theta_i = \frac{\angle_i * 180.0}{\pi} \tag{4.1}$$

donde $\angle_i$ denota al ángulo ϕ_i

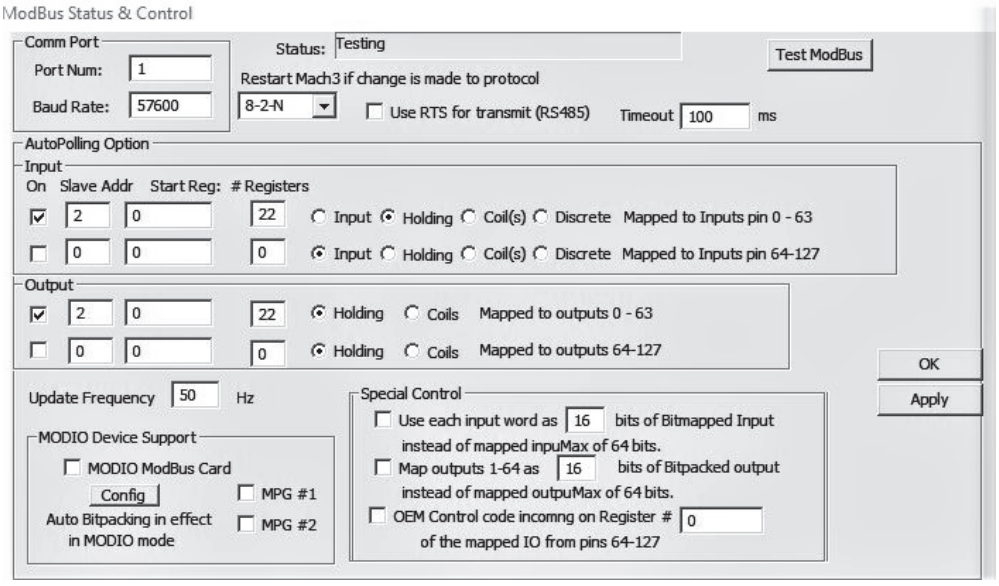


Figura 4.3: Configuración del protócolo Modbus en Mach3

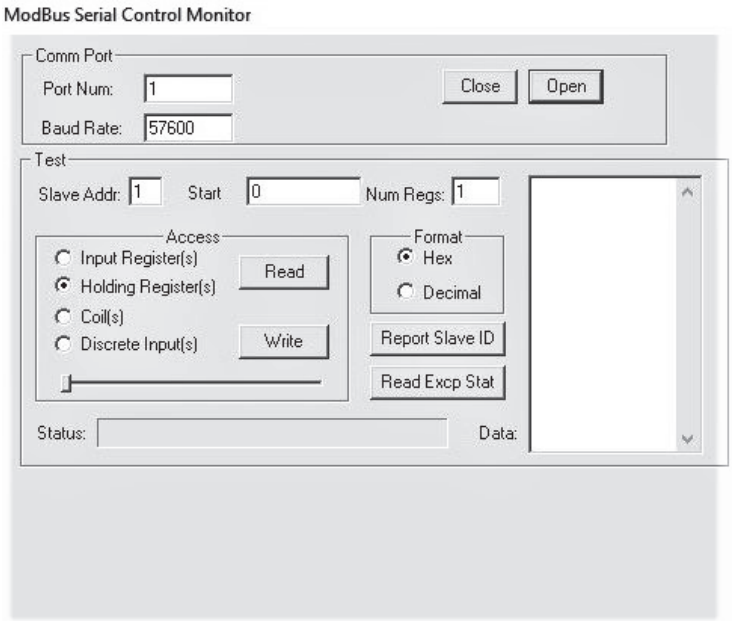


Figura 4.4: Prueba del protócolo Modbus en Mach3

Realizado lo anterior se procedió a hacer las conexiones entre los diferentes circuitos como se muestra en la Figura 4.5. Primeramente, se realizó un programa en donde se ingresa coordenadas en X, Y y Z y el robot llega al punto deseado, es decir, que hiciera la cinématica inversa. Esto lo realiza bien el robot, sin embargo, se observaron pequeños retardos en el accionamiento de los motores de las articulaciones 2 y 3. Estos retardos fueron más evidentes cuando se le cargó al microprocesador el código de seguimiento de trayectorias punto a punto. Al hacer pruebas de seguimiento de una línea recta y despúes un circulo se observó que el movimiento del robot no era fluido. Esto es debido a que no se tomó en cuenta el efecto de la gravedad, por ello se optó a modificar el programa y agregarle una ley de control donde se incluya este efecto. En este caso se trata de un control PD con compensación de gravedad.

Se eligió el control PD con compensación de gravedad por la factibilidad de su realización, ya que de la dinámica calculada anteriormente sólo se hace uso del vector de gravedad, además para el microprocesador del Arduino Mega 2560, el cálculo de posiciones demoraba en obtenerlos. Una vez implementado el programa con los valores de la Tabla 3.2, y tomando en cuenta la ecuación (3.39), sólo quedo sintonizar las ganacias K_p y K_d de manera experimental (prueba y

error) para tratar que el movimiento fuera suave. Se obtuvieron las constantes de ganancias de la Tabla 4.1.

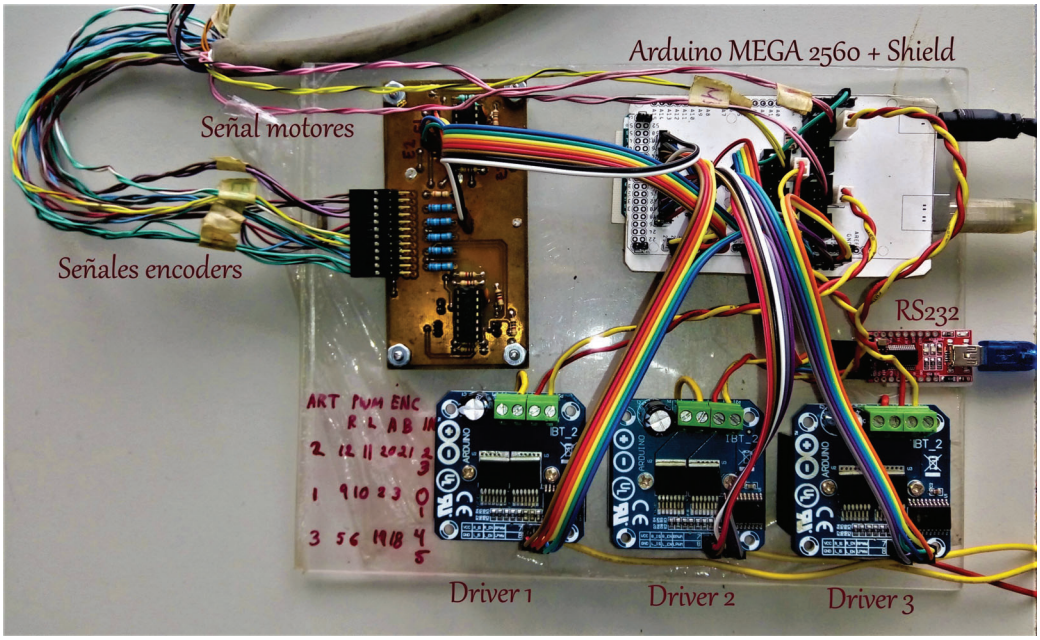


Figura 4.5: Conexiones entre los modulos de adquisición de datos y potencia para el robot académico SCORBOT ER-III

Ganancias			
Eslabón	K_p	K_d	K_i
1	6.0	0.1	0.7
2	0.05105	0.0021	
3	0.085	0.0028	

Tabla 4.1: Ganancias para el control de posición del robot SCORBOT ER-III.

4.1.2. Conversión del robot DENSO VP-6442G a máquina CNC

Para el robot académico DENSO VP-6442G, lo primero fue comprar el efector final, en este caso se adquirió un router de la marca Makita, después se procedió a diseñar el efector final en un software CAD, así como realizarlo en algún material rígido y de poco peso por las restricciones de carga que tiene el robot. Para esto se seleccionó una barra de aluminio para después realizar los cortes y los barrenos para la sujeción tanto del router como del robot. El diseño en CAD y en físico se muestra en la Figura 4.6. Los barrenos que presenta de más en la pieza física es porque el peso rebasaba el límite de 2.5 KG.

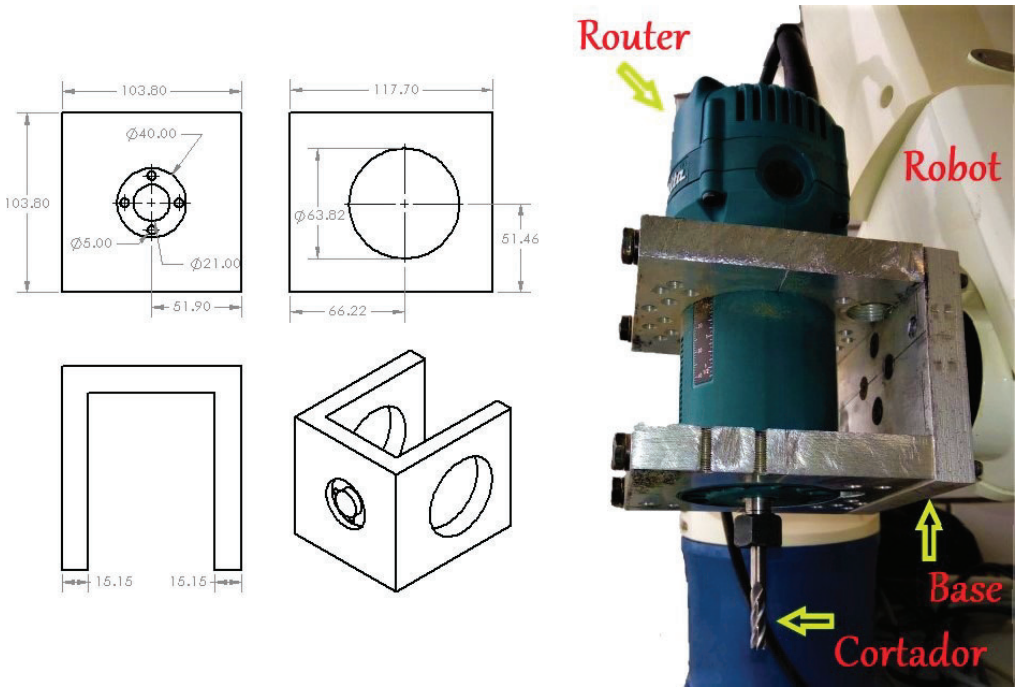


Figura 4.6: Base para el efector final del robot DENSO VP-6442G

Se realizaron diferentes pruebas haciendo piezas 3D mediante desbaste. Las primeras piezas realizadas fueron solo cortes en X, Y y Z, y en materiales como tablas de Fibras de Densidad Media (MDF, por sus siglas en inglés) y acrílico. Cabe resaltar que no se tenía conocimiento de los tipos de cortadores que se debe de utilizar en cada caso de material, para ello se requirió del apoyo de un mecánico

de la empresa CIA Robotics.

Modelo	G84304	G84308	G85008	C141716	55901TF	Unidad
Diametro de corte	1/16	1/8	1/8	1/4	1/4	in
Longitud total	1 1/2	2	2	2 1/2	6	in
Longitud de corte	1/16	1/8	5/32	3/4	1 1/2	in
Velocidad de avance	30	50	70	85	40	mm/s
Aveance de penetración	6	8	8	10	8	mm/s
Avance de retracción	10	12	12	12	12	mm/s
Cantidad de filos	2	2	4	4	4	
Profundidad de corte	1	1.5	1	2	2	mm
Ancho de corte	1/32	1/16	1/16	1/8	1/8	in

Tabla 4.2: Parámetros de cada herramienta de corte.

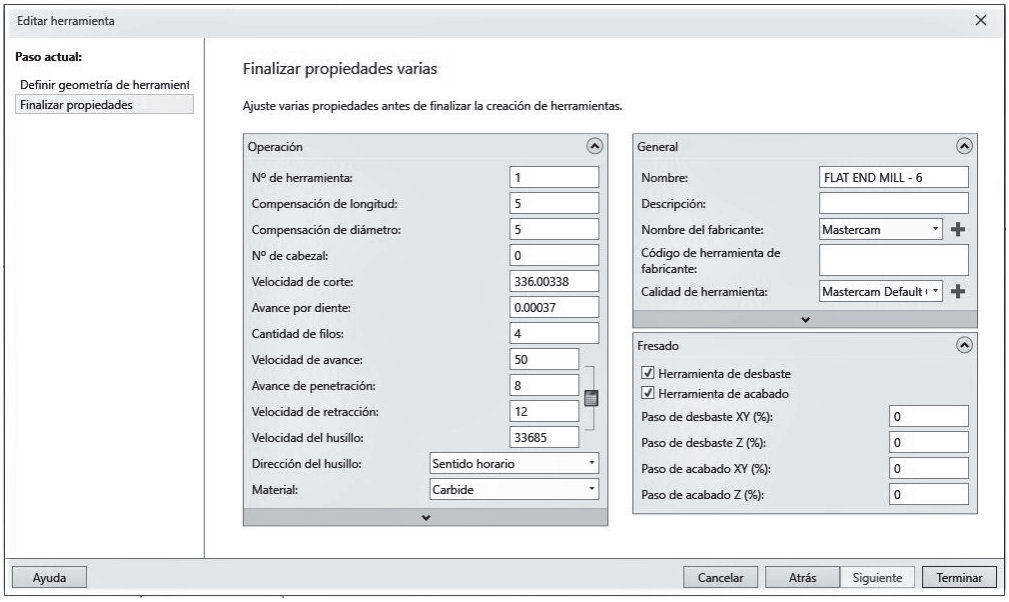


Figura 4.7: Modificación de parámetros de cortadores en Mastercam

Debido a las especificaciones de los materiales a cortar, del cortador y de que el router sólo tiene una velocidad, se realizó la conexión de un dimmer al router para poder variar la velocidad de giro. También se modificó la velocidad de avance, la velocidad de entrada, la velocidad de salida y la distancia de pasadas de corte, como se puede ver en la Figura 4.7. En la Tabla 4.2 se especifican estos parámetros para cada material y tipo de cortador, tanto de la hoja de datos de cada herramienta, así como de la experiencia del mecánico.

De las primeras piezas a realizarse fue el logo de la UACM, como se muestra en la Figura 4.8. En este diseño sólo se tenía contemplado realizar cortes en X, Y y profundidad Z. La simulación en los softwares Figura 4.9 y el generar el programa para Wincaps iii en RobotMaster no tuvo complicación alguna. En el software Wincaps iii solo se realizó los ajustes necesarios y se transfirio el archivo con extensión .PAC mediante conexión Ethernet entre el PC y el controlador.

Observando de la Figura 4.10 y los valores obtenidos de exactitud y presición de las Tablas 4.4 y 4.5 que, en X, Y y Z el robot trabajo bien, se tenía que realizar una pieza 3D más compleja, por lo que se realizó un engrane cónico. Esto con el objeto de ya no solo tener cortes en X, Y y Z, sino también tener cortes con orientación φ, ϑ, ψ, y ver cuáles son las limitaciones y alcances que se puede tener con un robot como CNC.

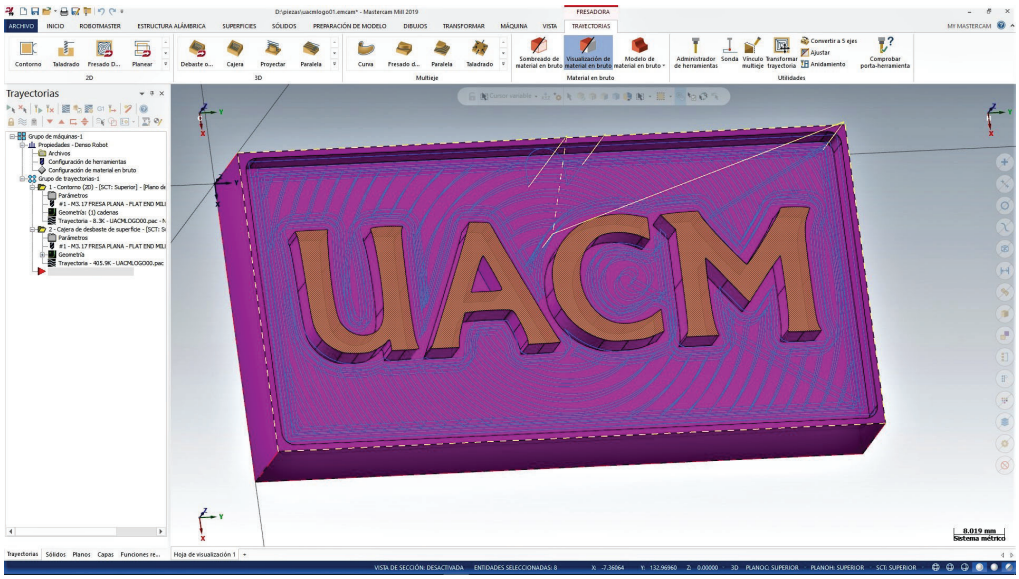


Figura 4.8: Pieza "logo UACM", en MasterCAM para el robot DENSO VP-6442G

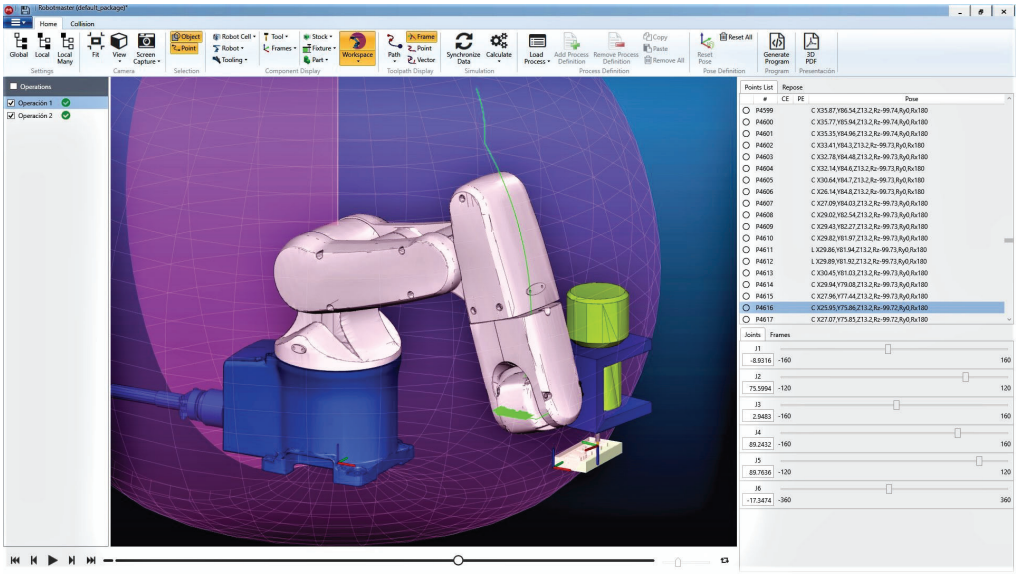


Figura 4.9: Simulación de mecanizado de la pieza “logo UACM”, en Robotmaster

El diseño de este engrane fue hecho de manera que el cortador realizará cortes orientados profundos, como tal no es una pieza replicada. El acomodar la pieza en RobotMaster resulta algo complicado, debido a que en ocasiones se marcan errores de límites de las articulaciones al tratar de cortar la pieza de un solo corte, esto porque en el ajuste del ángulo de orientación del efector final se mantenía en una posición fija, por lo que no se podía generar el código para el robot. La solución fue realizar la pieza en varios cortes. Lo siguiente fue la simulación que no presento errores y finalmente se generó el programa para el robot.

Porteriormente de generar el programa, se realizaron las modificaciones en el software Wincaps iii, tanto el ajuste de herramientas, la calibración del efector final y del área de trabajo [DR0 (2)]. Se cargo el archivo con extensión .PAC al controlador del robot académico DENSO mediante su conexión Ethernet. Teniendo esto listo se puso en marcha la tarea asignada al robot mediante la PC.

4.2. Resultados

4.2.1. Resultados de la conversión del robot SCORBOT ER-III como máquina CNC

Para el robot académico SCORBOT ER-III se limitaron los alcances propuestos, debido a que se planeaba implementarlo como CNC, es decir, que en el efector final contará con un herramental y se controlará su posición en el espacio 3D, pero debido al poco peso que logra soportar el robot (1 Kg) no se le pudo colocar tal herramental. Se tenía planeado un rectificador de la marca Dremel modelo 3000, pero su peso es aproximadamente 1.5 Kg y falta aún la estructura para sostenerlo. Por tanto, no se puede comprobar si las trayectorias que sigue son precisas.

El control de posición si lo realiza correctamente, esto se comprueba mediante el monitor serial. En el se muestra la posición deseada y la posición en la que se encuentra en el espacio 3D. De lo que se logró observar, las articulaciones 2 y 3 se activan al instante, con lo que puede decir que el control PD con compensación de gravedad está haciendo su tarea.

4.2.2. Resultados de la conversión del robot DENSO VP-6242G como máquina CNC

Para el robot académico DENSO VP-6442G, la primera prueba fue realizar el logo de la UACM en acrílico, como se muetra en la Figura 4.10. La realización de esta pieza fue algo tardado, debido a que aún no se contaba con cortadores de un diámetro mayor, sólo contaba con uno de 1/8". Una vez adquiridos varios, cada cortador debe de ser calibrado respecto a las posiciones de cada articulación del robot. El terminado es bueno, sin embargo, se observan errores en unas partes, estos son debido a que hace falta una mesa de coordenadas con la cual se pueda sujetar la pieza de manera que no se mueva.

La profundidad y la longitud de corte se midieron con un calibrador vernier como se muestra en la Tabla 4.3, pero, estos no concuerdan con el real debido a que el programa compensa los cortes con el diametro de la herramienta y como las

herramientas fueron las que tienen cono, entonces la compensación fue de 1/16". En Z no llego a la posición señalada por el hecho de que no se tiene aún una base fija, se trato de aproximar.



Figura 4.10: Pieza "logo UACM", realizado en acrílico con el robot DENSO VP-6442G como CNC

	Longitud de corte X	Longitud de corte Y	Profundidad de corte Z	Unidad
CAD	98.0 ±0.50	48.00 ±0.50	5.0 ±0.50	mm
Real	96.44	46.42	4.56	mm
Error de exactitud	1.59	3.29	8.8	%
Exactitud	98.41	96.71	91.21	%
Tiempo	21			min

Tabla 4.3: Medidas de corte de la pieza realizada "logo UACM".

La segunda prueba fue un engrane cónico, este sólo se realizó en MDF. En el MDF se observó que al realizar los cortes con orientaciones, estos se desviaban,

por ello se tuvo que calibrar el robot, el resultado obtenido se muestra en la Figura 4.11 y las mediciones con el calibrador vernier se presentan en la Tabla 4.4.

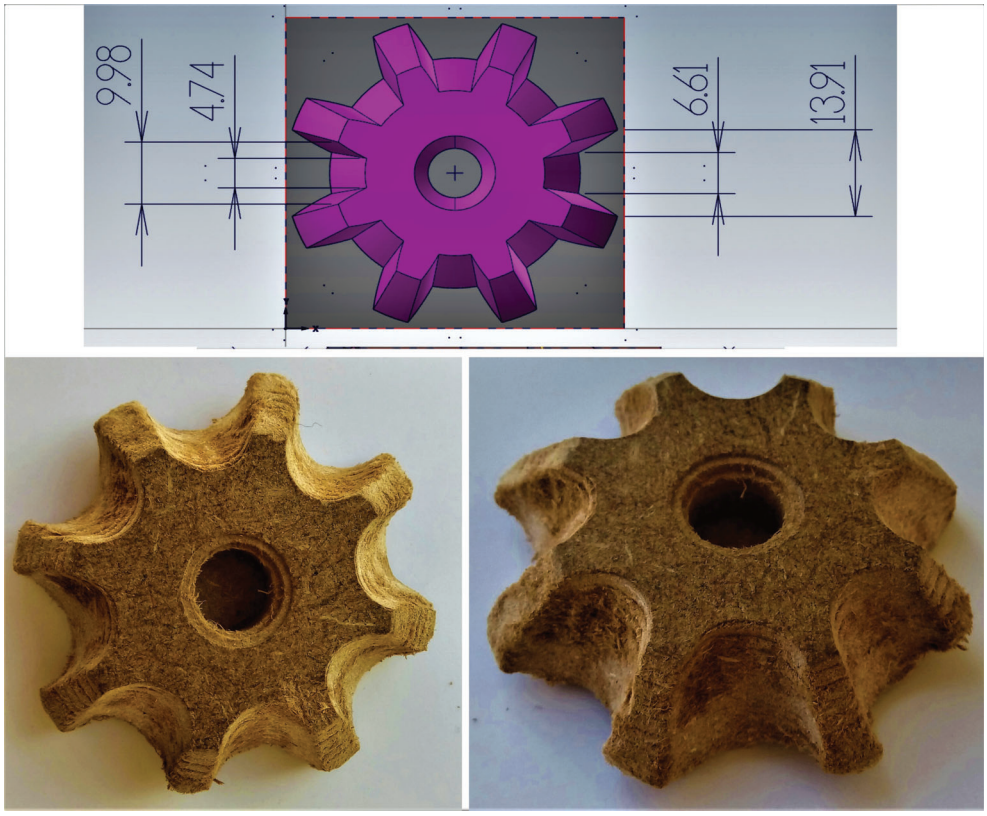


Figura 4.11: Pieza "Engrane cónico", realizada en MDF con el robot DENSO VP-6442G como CNC

Para saber la exactitud presentada en la Tabla 4.3 y la Tabla 4.4 de las piezas realizadas con el robot DENSO VP-6442G, se utilizan las siguientes formulas:

$$\%\ \text{Error de exactitud} = \left| \frac{\text{valor patron} - \ \text{valor medido}}{\text{valor patron}} \right| \times 100\,\% \tag{4.2a}$$

$$\%\ \text{Exactitud} = 100\,\% - \ \%\ \text{Error de exactitud} \tag{4.2b}$$

	Distancia 1	Distancia 2	Distancia 1	Distancia 2	Profundidad de corte Z	Unidad
CAD	6.61	4.74	9.98	13.91	7.00	mm
Real	6.73	4.58	10.13	13.93	6.94	mm
Error de exactitud	1.81	3.37	1.5	0.14	0.85	%
Exactitud	98.19	96.63	98.5	99.86	99.15	%
Tiempo	37					min

Tabla 4.4: Medidas de corte de la pieza realizada "engrane cónico".

y la presición de cada una de las piezas:

$$\%\ \text{Error de presición} = \left| \frac{\text{valor promedio} - \ \text{valor más lejano del valor promedio}}{\text{valor promedio}} \right| \times 100\,\% \tag{4.3a}$$

$$\%\ \text{Presición} = 100\,\% - \ \%\ \text{Error de presición} \tag{4.3b}$$

	valor promedio	valor más lejano	Error de presicion	Presicion
Pieza 1	95.44	91.21	4.43	96.57
Pieza 2	98.46	96.63	1.85	98.15

Tabla 4.5: Presición de las piezas realizadas con el robot DENSO VP-6442G.

4.3. Comparativa

En la Tabla 4.6 se presenta una comparativa de costos, beneficios y alcances con los dos robots como sistema CNC respecto a máquinas CNC.

Desde un punto de vista personal, dadas las aplicaciones industriales, son varias las ventajas al elegir un brazo robótico como máquina CNC a una máquina

CNC, pareciera contrario por los costos presentados en la Tabla 4.6, sin embargo, debido a que una máquina CNC está diseñada para una tarea específica, es decir, si su aplicación es desbaste, estará limitado a realizar sólo esa actividad y si se quiere por ejemplo cortar con láser, se tendría que comprar otro equipo, el costo aumenta al doble; en cambio un robot puede realizar una variedad de tareas, sólo cambiando el herramental del efector final y estará listo para otra aplicación, incluso operar como 3D printer.

	SCORBOT ER-III	DENSO VP-6442G	Máquina Router CNC
Precio	Uso S/control $10,000-$12,000 + Controlador $800 + Amplificadores $1,000 + Circuitos varios $200	Nuevo C/control $250,000 + Software $120,640 + efector final $2,000	Nuevo 3 DOF $16,000-$35,000 4 DOF $38,000-$80,000
Costo total	$12,000-$14,000	$372,640	$16,000-$80,000
Capacidad de carga	1.0 Kg	2.5 Kg	-
Ejes de trabajo	3	6	3 - 4

Tabla 4.6: Comparativa entre los diferentes robots y máquinas CNC.

Entre las ventajas de utilizar un robot como CNC a una máquina CNC están:

- El espacio de trabajo en una máquina CNC es pequeño, generalmente es un cubo pequeño, a comparación de un brazo robótico que tiene un gran área de trabajo.
- El lenguaje de programación en una máquina CNC es código G o código M. Un robot con el postprocesador puede convertir no sólo los 2 lenguajes de programación antes mencionados, sino cualquier otro lenguaje de programación.

- Los movimientos en una máquina CNC generalmente son de 3 a 4 DOF, en cambio con un robot se tienen hasta 6 DOF. con lo que se puede maquinar practicamente cualquier forma 3D.

Capítulo 5

Conclusiones y trabajos futuros

5.1. Conclusiones

Respecto al modelo cinemático del robot académico SCORBOT ER-III, que no se utilizó como tal para obtener la cinemática inversa, fue de gran utilidad para la realización del código para el microcontrolador, ya que, al tener el robot su posición inicial en coordenadas cartesianas $X = 44.0, Y = 0.0, Z = 0.0$, los ángulos de cada articulación deberían de ser $\theta_1 = 0.0, \theta_2 = -90.0, \theta_3 = 0.0$ y fueron correctos estos. Otro detalle que se observó fue que a medida que la articulación 2 se movía, la articulación 3 también lo hacía, es decir estas 2 articulaciones no están desacopladas. Razón por la cual en el programa se agrega un término de compensación a la articulación 3.

Para la obtención de la cinématica inversa del robot académico SCORBOT ER-III, por factibilidad se utilizó el método geométrico, ya que solo son 3 DOF. Si fuera el caso de un mayor número de DOF, es conveniente obtener la cinemática por el método algebraico. Sabiendo que el objetivo principal de la realización de este trabajo es utilizar el robot como CNC, con lo que una pieza de trabajo debe estar fija sobre una base, se optó por no agregar al programa del microcontrolador el caso de codo abajo, ya que en esta posición existen problemas de colisiones tanto con la estructura del robot, asi como la base para sujeción de la pieza de trabajo.

Los valores de las magnitudes físicas del modelo dinámico del robot académico SCORBOT ER-III, fueron basados en el artículo citado. De las matrices obtenidas, sólo se ocupo el vector de gravedad, dado que así está planteada la ley de control PD con efecto de gravedad.

Como se mencionó anteriormente, se hizo uso de la ley de control PD con compensación de gravedad debido a la factibilidad de este. Su implementación en el microcontrolador presentó detalles, dado que los movimientos que realizabán los motores de las articulaciones eran bruscos. Del modelo de un motor de corriente directa, se obtuvo la relación que existe entre el par-voltaje y los parámetros necesarios para hacer el cambio de una variable a otra. Implementando al microprocesador esta relación, los movimientos son mejores y se logra anular el problema del retraso en la activación de los motores de la articulación 2 y 3. Si el vector de gravedad está ausente en el programa para el microprocesador, el seguimiento de trayectoria no es posible en tiempo real.

La sintonización de las ganancias K_p y K_d no resulta sencillo, ya que a cada cambio de posición en coordenadas cartesianas la referencia cambia. Aunque la ley de control se simule en Matlab y parezca estable al seguimiento de una trayectoria, no funciona de la misma manera que ya implementado en el sistema físico.

Respecto al seguimiento de trayectoria, dado que es punto a punto y las coordenadas son muchas a cada segundo, al microprocesador le costaba realizar los cálculos, lo que se veía reflejado en que los movimientos que no eran del todo suaves. Se busco modificar la cantidad de datos que recibía el controlador de Mach3 pero no mejoraron los movimientos.

Para el robot académico DENSO VP-6442G, los resultados obtenidos fueron los esperados, como tal utilizar el robot como CNC, pero no únicamente en X, Y, Z, sino también con orientación, aunque se tenga que calibrar nuevamente cada cortador y cargar varias veces programas al controlador este robot hace su tarea correctamente.

Como conclusión general, basado en la Tabla 4.6, se puede afirmar que si es posible realizar un sistema de mecanizado CNC con un brazo robótico y pocos recursos, sin embargo, dependiendo la aplicación, la disponibilidad económica y que tanta exactitud sea necesaria, se elegiría un sistema u otro.

5.2. Mejoras y trabajos futuros

Entre las mejoras esta cambiar del Arduino Mega 2560 a un PLC, una DAQ o un FPGA, tomando en cuenta principalmente la lectura de encoders, la cantidad de salidas PWM, los protócolos de comunicaciones, en especial si cuenta con Mod-

bus y que tenga la posibilidad de enlazarlo con el software Matlab. Otro cambio sería el robot, la empresa CIA robotics, cuenta con otro robot más reciente de la misma marca SCORBOT modelo ER-VII que tiene mejores propiedades y las articulaciones si están desacopladas una de otra, pero cambiar de robot implica volver a realizar cálculos.

Para trabajo futuro está el implementar un mejor sistema de control de posición, donde se incluya aparte del vector de gravedad; la matriz de inercia, la matriz de Coriolis y el vector de fricción. También agregar un sistema de control de trayectorias con lo que se le restaría la comunicación con el software Mach 3 y utilizar la interfaz Simulink de Matlab, lo que facilitaría la parte del diseño del sistema de control.

Al robot DENSO VP-6442G implementado como CNC, agregarle un DOF más, que sería en la parte que sujeta la pieza de trabajo y con ello en un único programa se realizarían todos los cortes.

Por último, se tiene planeado que, a través de un escáner 3D, cualquier pieza pueda ser replicada con un robot como sistema de mecanizado CNC, pero para ello se ocupará un brazo robótico industrial profesional de la empresa CIA Robotics modelo ABB IRB-6600, con un alcance de aproximadamente 3 metros de altura y una capacidad de carga de 225 Kg.

Apéndice A

Datos técnicos, robot DENSO VP-6442G, SCORBOT ER-III

El Robot DENSO VP-6442G es un robot académico de 6 DOF ideal para intalaciones donde el espacio de operación es límitado. Sobrealen sus siguientes características [DR0 (3)].

- Excelente repetibilidad de ±0.02 mm
- Carga entre 2-2.5 kg
- Alcance de 420 mm
- Diseño ultracompacto
- Montaje en el piso o techo, sin necesidad de un software especial
- Excepcionalmente ligero con un peso entre los 13 a 17 kg
- Bajo consumo de energía, con todos los motores simultaneamente, ocupa un máximo de 300 W
- Cumple con normas ANSI y CE que le permiten su implementación global

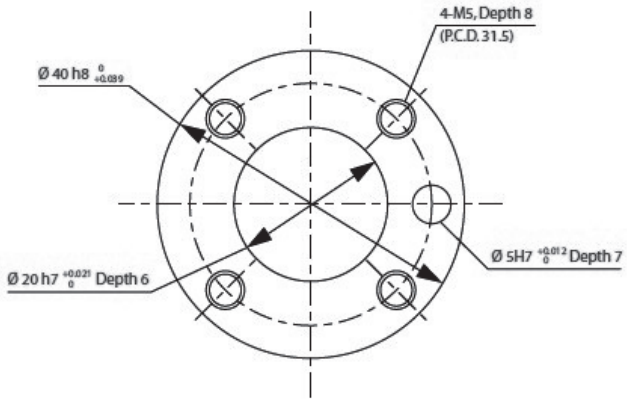


Figura A.1: Dimensiones para efector final del robot DENSO VP-6442G

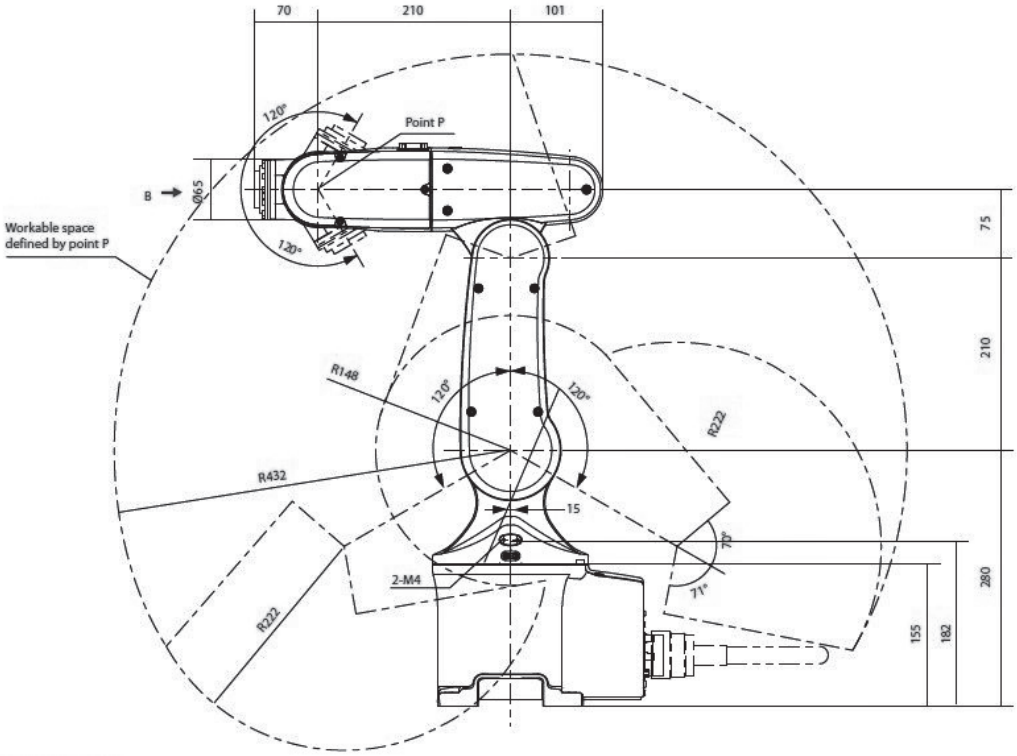


Figura A.2: Área de trabajo del robot DENSO VP-6442G. Vista lateral

		VP-6242G
Number of axes		6
Overall arm length (mm)		420
Second-arm width (mm)		112
Reach (to center of 5th axis, mm)		432
Minimum motion area (mm)		148
Maximum payload (kg)		2.0 (2.5 with flange pointing down)
Weight, approx. (kg)		15
Footprint (mm)		160 × 160
Cycle time (sec.)		0.99 (with 1-kg weight)
Maximum composite speed (mm/sec.)		3,900
Maximum joint speed (°/sec.)		J1: 250 J2: 187 J3: 250 J4, J5, J6: 300
Repeatability[1, 2] (mm)		±0.02
Max. allowable moment of inertia (kgm^2)	J4	0.03
	J5	
	J6	0.007
Range of motion (°)	J1	±160
	J2	±120
	J3	+160, +19
	J4	±160
	J5	±120
	J6	±360
Position detection		Absolute encoder
Drive motors and brakes		AC servomotors and brakes for all axes
Air piping (in-arm integrated)		4 systems (ø4 × 4)
Wiring (in-arm integrated)		9 lines (max. 1 A per line)
Air source	Operating pressure	1.0×10^5 Pa to 3.9×10^5 Pa
	Maximum allowable pressure	4.9×10^5 Pa
Protection class		IP30
Ceiling-mount		✓

Figura A.3: Características generales del robot DENSO VP-6242G

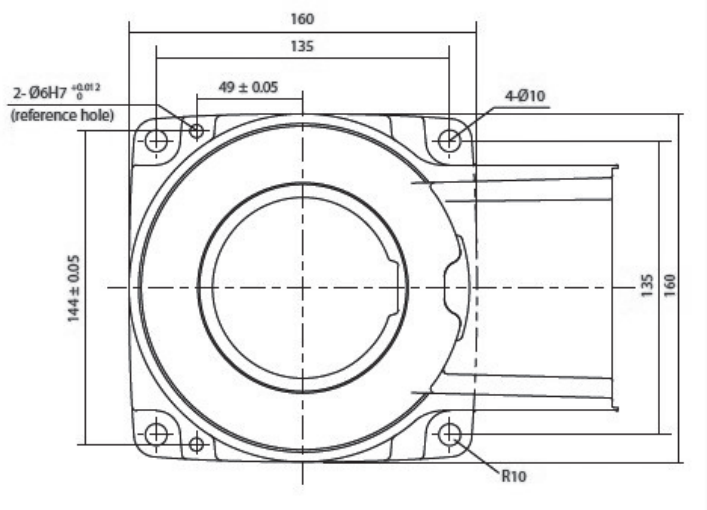


Figura A.4: Dimensiones de la base del robot DENSO VP-6442G

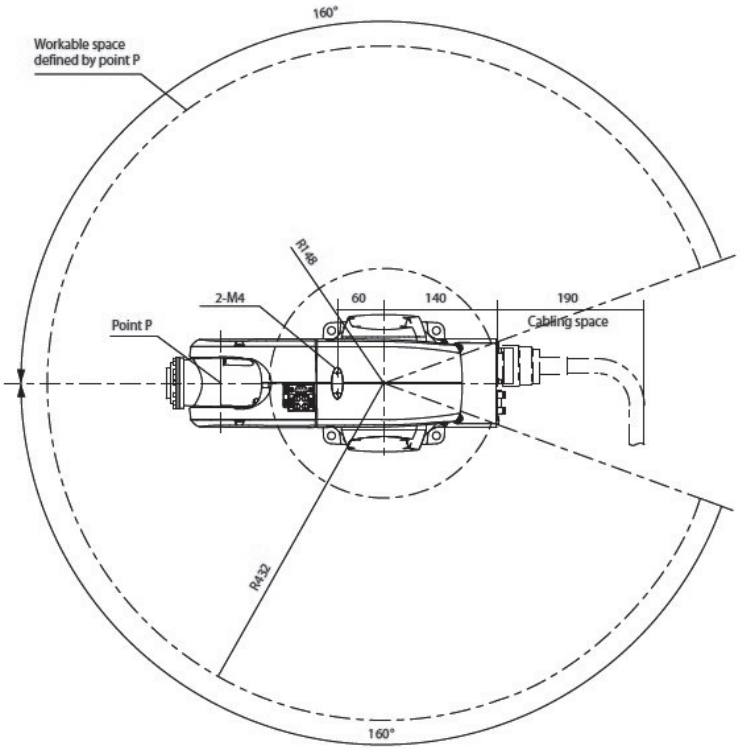


Figura A.5: Área de trabajo del robot DENSO VP-6442G. Vista superior

El robot SCORBOT ER-III es un brazo robótico con 5 DOF de uso didáctico [ER0 (1)].

Item	Specification
Mechanical Structure	Vertically articulated 5 axes plus gripper Control of 8 axes simultaneously
Working Envelope:	
Axis 1: Base Rotation	310°
Axis 2: Shoulder Rotation	+ 130° / –35°
Axis 3: Elbow Rotation	± 130°
Axis 4: Wrist Pitch	± 130°
Axis 5: Wrist Roll	Unlimited
Maximum Working Radius	610 mm (24.4")
Gripper Opening	75 mm (3") without rubber pads 65 mm (2.56") with rubber pads
Maximum Work Load	1 kg (2.2 lb.)
Transmission	Gears, timing belts and lead screw
Actuators	6 DC servo motors with closed-loop control
Feedback	Optical encoders on all axes
Hard Home	Fixed reference position on all axes
Repeatability	± 0.5 mm (± 0.02")
Maximum Speed	330 mm/sec. (13"/sec.)
Weight	
Robot Arm	11 kg (24 lb.)
Controller	5 kg (11 lb.)

Figura A.6: Características generales del robot SCORBOT ER-III

Apéndice B

Programas

B.1. Programa en lenguaje C para el microcontrolador ATMEGA 2560

```
//SCORBOT ER-III

#include <avr/io.h>
#include <util/delay.h>
#include <math.h>
#include <SimpleModbusSlave.h>
const int HOLDING_REGS_SIZE = 22; // 11 registros envian informacion a Mach3
                                  //  y depues 11 registros reciben informacion de
Mach3
unsigned int holdingRegs[HOLDING_REGS_SIZE]; // function 3 and 16 register array
int rcd[11];                      //montiene el envio de informacion
int mcd[11];                      //mantiene la recepcion de informacion

/////////////// Configuraciones para los PWM ////////////////////
double ADC_SetPoint1      = 0;
double ADC_SetPointOld1   = 0;
double ADC_ServoPoti1     = 0;
double ADC_ServoPotiOld1  = 0;
double dutyCycle1         = 150;
double ADCdiff1           = 0;
double timeDiff1          = 0;

double ADC_SetPoint2      = 0;
double ADC_SetPointOld2   = 0;
double ADC_ServoPoti2     = 0;
double ADC_ServoPotiOld2  = 0;
double U2                 = 150.0;
double ADCdiff2           = 0;
double timeDiff2          = 0;

double ADC_SetPoint3      = 0;
double ADC_SetPointOld3   = 0;
double ADC_ServoPoti3     = 0;
double ADC_ServoPotiOld3  = 0;
double U3                 = 150.0;
double ADCdiff3           = 0;
double timeDiff3          = 0;
```

```
/////////////// Configuraciones para los calculos ////////////////

double time;
double rxy    = 0.0;
double the1   = 0.0;
double the2   = 0.0;
double alpha  = 0.0;
double beta   = 0.0;
double ar     = 0.0;
double G2q    = 0.0;
double G3q    = 0.0;
double T2     = 0.0;
double T21    = 0.0;
double T22    = 0.0;
double T31    = 0.0;
double T32    = 0.0;
double T3     = 0.0;
double g      = 9.81;
double Ts     = 0.1;
double q2e    = 0.0;
double q3e    = 0.0;
double qp2    = 0.0;
double qp3    = 0.0;

float pi      = 3.14159;
float l1      =0.220;
float l2      =0.220;
float l3      =0.150;
float Ka      =15.0;
float r       =177.0;
float Ra      =3.0;
float m2      =2.2483;
float m3      =1.957;
float mf      =1.0;
float lc2     =0.052;
float lc3     =0.1376;
float Kp2     =0.05105;
float Kd2     =0.0021;
float Kp3     =0.085;
float Kd3     =0.0028;

////////////// Configuraciones para los encoders ///////////////////

long count1,count2,count3; // contador para calculo de posicion
boolean A,B,C,D,E,F;
double posicion1, error1, posicion2, error2,posicion3, error3;
byte state, statep, indexx,state2, statep2, indexx2,state3, statep3, indexx3;
int QEM[16]= {0,-1,0,1,1,0,-1,0,0,1,0,-1,-1,0,1,0};

///// Definiciones de constantes //////

#define P_FRACTION           6.0
#define I_FRACTION           0.7
#define D_FRACTION           0.1
#define V_WINDOW             0.18
#define MIN_DUTYCYCLE        10        //0 - 255 (25)
#define MAX_DUTYCYCLE        255       //0 - 255 (255)
#define SOFT_START           0.3       //0.00 - 1.00 (0.30) 1.00 = OFF
#define D_FRACTION_DEMO1     0
#define EMERGENCY_SHUTDOWN   4
#define SHOOT_THROUGH_PAUSE  10

#define V_WINDOW2              0.18
#define MIN_DUTYCYCLE2         10
#define MAX_DUTYCYCLE2         255
#define SOFT_START2            0.3
#define D_FRACTION_DEMO2       0
#define EMERGENCY_SHUTDOWN2    4
#define SHOOT_THROUGH_PAUSE2   10

#define V_WINDOW3              0.15
#define MIN_DUTYCYCLE3         10
#define MAX_DUTYCYCLE3         255
#define SOFT_START3            0.3
#define D_FRACTION_DEMO3       0
#define EMERGENCY_SHUTDOWN3    4
#define SHOOT_THROUGH_PAUSE3   10
```

```
////// Definicion de pines /////////
#define SERVO_PWM_R1 9            //Right PWM pin BTS7960 bridge
#define SERVO_PWM_L1 10           //Left PWM pin BTS7960 bridge
#define SERVO_PWM_R2 12           //Right PWM pin BTS7960 bridge
#define SERVO_PWM_L2 11           //Left PWM pin BTS7960 bridge
#define SERVO_PWM_R3 6            //Right PWM pin BTS7960 bridge
#define SERVO_PWM_L3 5            //Left PWM pin BTS7960 bridge
#define SERVO_SENSOR A2           //Analog input servo sensor
#define POTI A3                   //Analog input potentiometer

void setup(){

  Serial2.begin(115200);
  pinMode(13, OUTPUT);                    // Indicador LED
  pinMode(2,INPUT);                       // Pines interrupciones
  pinMode(3,INPUT);
  pinMode(19,INPUT);
  pinMode(18,INPUT);
  pinMode(21,INPUT);
  pinMode(20,INPUT);
  attachInterrupt(2,Achange,CHANGE);      // No. de interrupcion
  attachInterrupt(3,Bchange,CHANGE);
  attachInterrupt(0,Cchange,CHANGE);
  attachInterrupt(1,Dchange,CHANGE);
  attachInterrupt(4,Echange,CHANGE);
  attachInterrupt(5,Fchange,CHANGE);
  pinMode(SERVO_PWM_R1, OUTPUT);
  pinMode(SERVO_PWM_L1, OUTPUT);
  pinMode(SERVO_PWM_R2, OUTPUT);
  pinMode(SERVO_PWM_L2, OUTPUT);
  pinMode(SERVO_PWM_R3, OUTPUT);
  pinMode(SERVO_PWM_L3, OUTPUT);
  pinMode(SERVO_SENSOR, INPUT);
  pinMode(POTI, INPUT);
  digitalWrite(4, HIGH);
  digitalWrite(7, HIGH);
  digitalWrite(8, HIGH);
  modbus_configure(&Serial, 115200, SERIAL_8N2, 2, 13, HOLDING_REGS_SIZE, holdingRegs);
}

void loop(){

////// Codigo para comunicar con Mach3 /////////

  if(digitalRead(4) == 1) bitSet(rcd[0],0);
  else bitClear(rcd[0],0);

  if(digitalRead(7) == 1) bitSet(rcd[0],1);
  else bitClear(rcd[0],1);

  if(digitalRead(8) == 1) bitSet(rcd[0],2);
  else bitClear(rcd[0],2);

  rcd[3] = analogRead(A0);
  rcd[4] = analogRead(A1);

  //Otras variables:
  rcd[1] = 0x1111;
  rcd[2] = 0x2222;
  rcd[5] = 0x3333;
  rcd[6] = 32767;
  rcd[7] = 12345;
  rcd[8] = 54321;
  rcd[9] = 0;
  rcd[10] = 1;

  for (int i=0; i < 11; i++)
    holdingRegs[i] = rcd[i];

  modbus_update();

  for (int i=0; i < 11; i++)
    mcd[i] = holdingRegs[i+11];
```

```
/////// Posiciones recibidas de Mach3 ////////
    float Xpos = (mcd[2]+(mcd[3]/1000.00))/100.0;
    float Ypos = (mcd[4]+(mcd[5]/1000.00))/100.0;
    float Zpos = (mcd[6]+(mcd[7]/1000.00))/100.0;

///// Calculos de cinemetica inversa ////////
    double R=sqrt(pow(Xpos,2)+pow(Ypos,2)+pow(Zpos,2));
    double L=l1+l2+l3;

      while (R > L)
      {
        Serial.print("Posicion fuera de alcance");
      }

      if (Xpos == 0)
      {
        if (Ypos < 0)
        {
          rxy=Ypos*(-1);
        }

        rxy = Ypos;
        the1 = asin ((double)(Ypos/rxy));
        ar = sqrt(pow(Ypos,2)+pow(Zpos,2));
        beta = atan((double)(Zpos/Ypos));

      }

      else
      {
        the1 = atan((double)(Ypos/Xpos));
        rxy = Xpos/cos((double)(the1));
        ar = sqrt(pow(rxy,2)+pow(Zpos,2));
        beta = atan((double)(Zpos/rxy));
      }

  double alpha = acos((double)(pow(l1,2)+pow(ar,2)-pow(l2,2))/(2*l1*ar));
  double the2 = beta+alpha;

      if (Xpos<0)
      {
        the2 = beta+alpha+180;
      }

  double f = l1*sin(the2);
  double h = Zpos-f;
  double mu = asin((double)(h/l2));
  double the3 = (the2-mu);

////// Conversion a grados ///////
  double theta1 = (the1*180)/pi;
  double theta2 = (the2*180)/pi;
  double theta3 = (the3*180)/pi;

  Serial2.print("theta1: ");
  Serial2.print(theta1);
  Serial2.print("theta2: ");
  Serial2.print(theta2);
  Serial2.print("theta3: ");
  Serial2.println(theta3);
  Serial2.print("X: ");
  Serial2.print(Xpos);
  Serial2.print("Y: ");
  Serial2.print(Ypos);
  Serial2.print("Z: ");
  Serial2.println(Zpos);
  Serial2.print("posicion1: ");
  Serial2.print(posicion1);
  Serial2.print("posicion2: ");
  Serial2.print(posicion2);
  Serial2.print("posicion3: ");
  Serial2.println(posicion3);
  Serial2.println("  ");
```

```
///// Control PID primer articulacion ////////

  posicion1= count1*(360.0/16600.0);
  ADC_ServoPoti1 = posicion1;
  ADC_SetPoint1 = theta1;
  ADCdiff1 = ADC_SetPoint1 - ADC_ServoPoti1;
  dutyCycle1 = abs(ADCdiff1) * P_FRACTION;
  dutyCycle1 += timeDiff1 * I_FRACTION;
  dutyCycle1 += abs(ADC_SetPointOld1 - ADC_SetPoint1) * D_FRACTION;
  timeDiff1++;

     if(abs(ADCdiff1) < V_WINDOW){
      dutyCycle1 = 0;
      timeDiff1 = 0;
    }

    if(abs(ADC_ServoPotiOld1 - ADC_ServoPoti1) < EMERGENCY_SHUTDOWN && dutyCycle1
                == MAX_DUTYCYCLE && timeDiff1 > 50){
      analogWrite(SERVO_PWM_L1, 0);
      delayMicroseconds(SHOOT_THROUGH_PAUSE);
      digitalWrite(SERVO_PWM_R1, 0);
      delay(1000);
      timeDiff1 = 0;
    }
    else{
      if(ADCdiff1 > 0){
        analogWrite(SERVO_PWM_L1, 0);
        delayMicroseconds(SHOOT_THROUGH_PAUSE);
        digitalWrite(SERVO_PWM_R1, 0);
        delayMicroseconds(SHOOT_THROUGH_PAUSE);
        analogWrite(SERVO_PWM_L1, dutyCycle1);
      }
      if(ADCdiff1 < 0){
        analogWrite(SERVO_PWM_L1, 0);
        delayMicroseconds(SHOOT_THROUGH_PAUSE);
        digitalWrite(SERVO_PWM_R1, 1);
        delayMicroseconds(SHOOT_THROUGH_PAUSE);
        analogWrite(SERVO_PWM_L1, 255 - dutyCycle1);
      }
    }

///// Control PD segunda articulacion /////////

  posicion2= count2*(180.0/6200.0);
  ADC_ServoPoti2 = posicion2;
  ADC_SetPoint2 = theta2;
  qp2 = abs(ADC_ServoPoti2-ADC_ServoPotiOld2)/Ts;
  G2q = m2*g*lc2*cos(theta2)+m3*g*l2*cos(theta2)+m3*g*lc3*cos(theta2+theta3)
              +mf*g*l2*cos(theta2)+mf*g*l3*cos(theta2+theta3);
  q2e = ADC_SetPoint2-ADC_ServoPoti2;
  T2  = abs(q2e) * Kp2;
  T2 += (qp2 * Kd2);
  T2 -= G2q;
  U2  = abs(T2*r*Ra/Ka);
  timeDiff2++;

    if(abs(q2e) < V_WINDOW2){
      U2 = 0;
      timeDiff2 = 0;
    }

    if(abs(ADC_ServoPotiOld2 - ADC_ServoPoti2) < EMERGENCY_SHUTDOWN2 && U2
        == MAX_DUTYCYCLE2 && timeDiff2 > 50){
      analogWrite(SERVO_PWM_L2, 0);
      delayMicroseconds(SHOOT_THROUGH_PAUSE2);
      digitalWrite(SERVO_PWM_R2, 0);
      delay(1000);
      timeDiff2 = 0;
    }
    else{
      if(q2e > 0){
        analogWrite(SERVO_PWM_L2, 0);
        delayMicroseconds(SHOOT_THROUGH_PAUSE2);
        digitalWrite(SERVO_PWM_R2, 0);
        delayMicroseconds(SHOOT_THROUGH_PAUSE2);
        analogWrite(SERVO_PWM_L2, U2);
```

```
      }
      if(q2e < 0){
        analogWrite(SERVO_PWM_L2, 0);
        delayMicroseconds(SHOOT_THROUGH_PAUSE2);
        digitalWrite(SERVO_PWM_R2, 1);
        delayMicroseconds(SHOOT_THROUGH_PAUSE2);
        analogWrite(SERVO_PWM_L2, 255 - U2);
      }
    }

///// Control PD tercera articulacion /////////

  posicion3= count3*(180.0/6200.0);
  ADC_ServoPoti3 = posicion3;
  ADC_SetPoint3 = theta3-theta2;
  qp3 = abs(ADC_ServoPoti3-ADC_ServoPotiOld3)/Ts;
  G3q = m3*g*lc3*cos(theta2+theta3)+mf*g*l3*cos(theta2+theta3);
  q3e = ADC_SetPoint3-ADC_ServoPoti3;
  T3  = abs(q3e) * Kp3;
  T3 += qp3 * Kd3;
  T3 -= G3q;
  U3  = abs(T3*r*Ra/Ka);
  timeDiff3++;

    if(abs(q3e) < V_WINDOW3){
      U3 = 0;
      timeDiff3 = 0;
    }

    if(abs(ADC_ServoPotiOld3 - ADC_ServoPoti3) < EMERGENCY_SHUTDOWN3 && U3
       == MAX_DUTYCYCLE3 && timeDiff3 > 50){
      analogWrite(SERVO_PWM_L3, 0);
      delayMicroseconds(SHOOT_THROUGH_PAUSE3);
      digitalWrite(SERVO_PWM_R3, 0);
      delay(1000);
      timeDiff3 = 0;
    }
    else{
      if(q3e > 0){
        analogWrite(SERVO_PWM_L3, 0);
        delayMicroseconds(SHOOT_THROUGH_PAUSE3);
        digitalWrite(SERVO_PWM_R3, 0);
        delayMicroseconds(SHOOT_THROUGH_PAUSE3);
        analogWrite(SERVO_PWM_L3, U3);
      }
      if(q3e < 0){
        analogWrite(SERVO_PWM_L3, 0);
        delayMicroseconds(SHOOT_THROUGH_PAUSE3);
        digitalWrite(SERVO_PWM_R3, 1);
        delayMicroseconds(SHOOT_THROUGH_PAUSE3);
        analogWrite(SERVO_PWM_L3, 255 - U3);
      }
    }

/////// Actualizacion de variables ////////

  ADC_SetPointOld1 = ADC_SetPoint1;
  ADC_ServoPotiOld1 = ADC_ServoPoti1;

  ADC_SetPointOld2 = ADC_SetPoint2;
  ADC_ServoPotiOld2 = ADC_ServoPoti2;

  ADC_SetPointOld3 = ADC_SetPoint3;
  ADC_ServoPotiOld3 = ADC_ServoPoti3;

  delay(15);
}

///// Lectura de encoders por interrupciones ////////

void Achange()
{
  A = digitalRead(20);
  B = digitalRead(21);
  if ((A == HIGH)&&(B == HIGH))state = 0;
  if ((A == HIGH)&&(B == LOW))state = 1;
  if ((A == LOW)&&(B == LOW))state = 2;
```

```
  if ((A == LOW)&&(B == HIGH))state = 3;
  indexx = 4*state + statep;
  count1 = count1 + QEM[indexx];
  statep = state;
}

void Bchange()
{
  A = digitalRead(20);
  B=  digitalRead(21);
  if ((A == HIGH)&&(B == HIGH))state = 0;
  if ((A == HIGH)&&(B == LOW))state = 1;
  if ((A == LOW)&&(B == LOW))state = 2;
  if ((A == LOW)&&(B == HIGH))state = 3;
  indexx = 4*state + statep;
   count1 = count1 + QEM[indexx];
   statep = state;
}

void Cchange()
{
  C = digitalRead(3);
  D=  digitalRead(2);
  if ((C == HIGH)&&(D == HIGH))state2 = 0;
  if ((C == HIGH)&&(D == LOW))state2 = 1;
  if ((C == LOW)&&(D == LOW))state2 = 2;
  if ((C == LOW)&&(D == HIGH))state2 = 3;
  indexx2 = 4*state2 + statep2;
  count2 = count2 + QEM[indexx2];
  statep2 = state2;
}

void Dchange()
{
  C = digitalRead(3);
  D=  digitalRead(2);
  if ((C == HIGH)&&(D == HIGH))state2 = 0;
  if ((C == HIGH)&&(D == LOW))state2 = 1;
  if ((C == LOW)&&(D == LOW))state2 = 2;
  if ((C == LOW)&&(D == HIGH))state2 = 3;
  indexx2 = 4*state2 + statep2;
   count2 = count2 + QEM[indexx2];
   statep2 = state2;
}

void Echange()
{
  E = digitalRead(18);
  F = digitalRead(19);
  if ((E == HIGH)&&(F == HIGH))state3 = 0;
  if ((E == HIGH)&&(F == LOW))state3 = 1;
  if ((E == LOW)&&(F == LOW))state3 = 2;
  if ((E == LOW)&&(F == HIGH))state3 = 3;
  indexx3 = 4*state3 + statep3;
  count3 = count3 + QEM[indexx3];
  statep3 = state3;
}

void Fchange()
{
  E = digitalRead(18);
  F = digitalRead(19);
  if ((E == HIGH)&&(F == HIGH))state3 = 0;
  if ((E == HIGH)&&(F == LOW))state3 = 1;
  if ((E == LOW)&&(F == LOW))state3 = 2;
  if ((E == LOW)&&(F == HIGH))state3 = 3;
  indexx3 = 4*state3 + statep3;
   count3 = count3 + QEM[indexx3];
   statep3 = state3;
}
```

B.2. Programas en lenguaje G

A continuación se muestran 2 programas en lenguaje G, el primero la trayectoria es una línea y el segundo un círculo.

```
N0010 (Filename: line1.tap)
N0020 (Post processor: Mach3.scpost)
N0030 (Date:10/10/2019 Time:13:02:57)
N0040 G21 (Units: Metric)
N0050 G40 G90 G91.1
N0060 F1
N0070 (Part: line1)
N0080 (Operation: Compensacion exterior, Color 0, T1: Fresadora/Router,
       4 mm Diametro, 1 mm Profundidad)
N0090 S1000 (Fresadora/Router, 4 mm Diametro)
N0100 T1 M06
N0110  G43 H1
N0120 G00 Z4.0000
N0130 S1000 M03
N0140 X25.0000 Y8.2828
N0150 Z0.5000
N0160 G01 Z-1.000 F30.0
N0170 Y0.0000 F40.0
N0180 Y-11.7172
N0190 G00 Z4.0000
N0200 M05
N0210 M05 M30
```

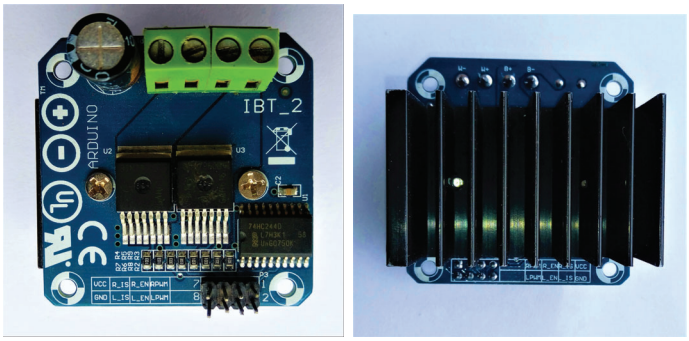
```
N0010 (Filename: circulo1.tap)
N0020 (Post processor: Mach3.scpost)
N0030 (Date:04/10/2019 Time:17:23:43)
N0040 G21 (Units: Metric)
N0050 G40 G90 G91.1
N0060 F1
N0070 (Part: circle2)
N0080 (Operation: Compensacion interior, Color 0, T1: Fresadora/Router,
       4 mm Diametro, 1 mm Profundidad)
N0090 S1000 (Fresadora/Router, 4 mm Diametro)
N0100 T1 M06
N0110  G43 H1
N0120 G00 Z4.0000
N0130 S1000 M03
N0140 X23.9142 Y-1.4142
N0150 Z0.5000
N0160 G01 Z-1.000 F180.0
N0170 X22.5000 Y-0.0000 F100.0
N0180 G02 X22.5000 Y-0.0000 Z-1.0000 I7.5000 J-0.0000
N0190 G01 X23.9142 Y1.4142 Z-1.000
N0200 G00 Z4.0000
N0210 M05
N0220 M05 M30
```

Apéndice C

Circuitos electrónicos

C.1. Driver para motor puente H, BTS7960

El driver BTS7960 es un modulo de alta potencia para controlar el sentido de giro de un motor de DC, mediante PWM.



(a) ***Vista frontal*** **(b)** ***Vista trasera***

Figura C.1: Modulo driver BTS7960

Sus características principales son las siguientes:

- Voltaje de entrada : 6-27 VDC
- Corriente máxima : 43 A
- Capacidad PWM : 25 kHz
- Nivel de entrada para control : 3.3-5 VDC
- Ciclo de trabajo : 0-100 %
- Bloqueo por sobre voltaje
- Apagado por bajo voltaje

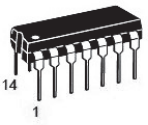
C.2. IC Schmitt Trigger modelo 74ls14n

El IC Schmitt Trigger 74ls14n cuenta con 14 pines, 2 de alimentación, 6 de entradas y 6 de salidas negadas. La principal función de este circuito es evitar el ruido de señales, esto haciendo uso de histeresis como se ve en la Figura C.4.

		SN5414			SN7414			UNIT
		MIN	NOM	MAX	MIN	NOM	MAX	
V_{CC}	Supply voltage	4.5	5	5.5	4.75	5	5.25	V
I_{OH}	High-level output current			–0.8			–0.8	mA
I_{OL}	Low-level output current			16			16	mA
T_A	Operating free-air temperature	–55		125	0		70	°C

PARAMETER	TEST CONDITIONS‡			SN7414			UNIT
				MIN	TYP§	MAX	
V_{T+}	V_{CC} = 5 V			1.5	1.7	2	V
V_{T-}	V_{CC} = 5 V			0.6	0.9	1.1	V
Hysteresis ($V_{T+} - V_{T-}$)	V_{CC} = 5 V			0.4	0.8		V
V_{IK}	V_{CC} = MIN,	I_I = –12 mA				–1.5	V
V_{OH}	V_{CC} = MIN,	V_I = 0.6 V,	I_{OH} = –0.8 mA	2.4	3.4		V
V_{OL}	V_{CC} = MIN,	V_I = 2 V,	I_{OL} = 16 mA		0.2	0.4	V
I_{T+}	V_{CC} = 5 V,	$V_I = V_{T+}$			–0.43		mA
I_{T-}	V_{CC} = 5 V,	$V_I = V_{T-}$			–0.56		mA
I_I	V_{CC} = MAX,	V_I = 5.5 V				1	mA
I_{IH}	V_{CC} = MAX,	V_{IH} = 2.4 V				40	µA
I_{IL}	V_{CC} = MAX,	V_{IL} = 0.4 V			–0.8	–1.2	mA
I_{OS}¶	V_{CC} = MAX			–18		–55	mA
I_{CCH}	V_{CC} = MAX				22	36	mA
I_{CCL}	V_{CC} = MAX				39	60	mA

Figura C.2: Características generales en DC del IC 74ls14n



N SUFFIX
PLASTIC
CASE 646-06

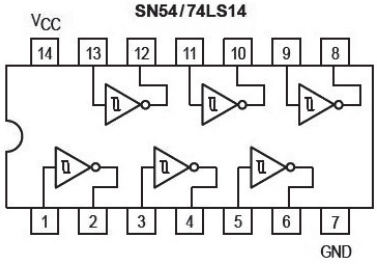


Figura C.3: Diagrama esquemático y pinout del IC 74ls14n

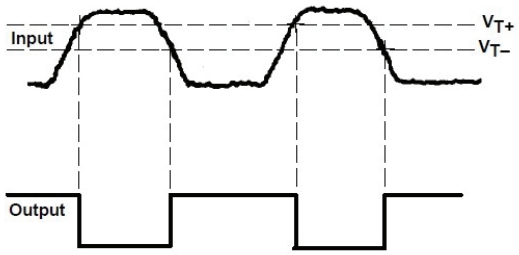


Figura C.4: Conformación del pulso del IC 74ls14n

C.3. Convertidor USB a TTL FT232RL

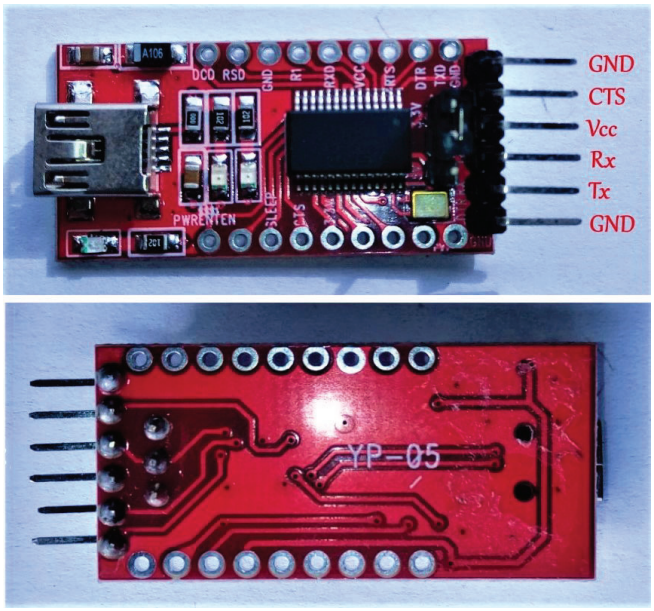


Figura C.5: Modulo USB a TTL FT232RL

Los usos generales de este modulo son las siguientes:

- Programación de microprocesadores como ARM, AVR, MICROCHIP, etc.
- Comunicación con muchos dispositivos.
- Vcc. Salida de voltaje positivo, tanto 3.3 o 5 VDC. Manejan un aproximado de 50 mA y 500 mA respectivamente.

- Rx. Pin de recepción de datos por serial.
- Tx. Pin de transmición de datos por serial.
- CTS. Control de flujo.
- Transferencia de datos de manera asíncrona.
- Cuenta con su cristal.
- Velocidad de transferencia de datos desde 300 a 3M Baudios.

C.4. Arduino Mega 2560

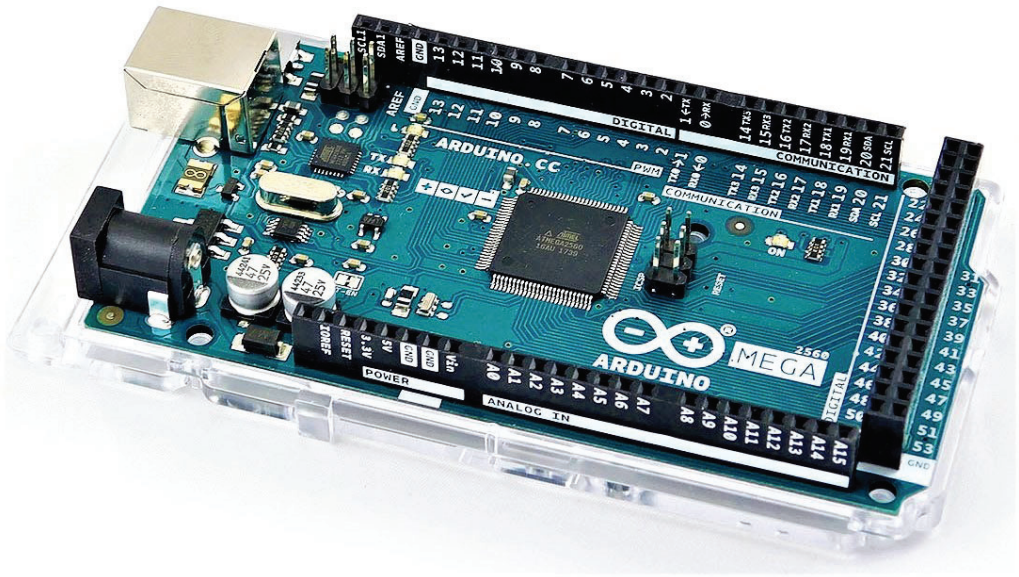


Figura C.6: Arduino Mega 2560

El Arduino Mega 2560 es un microcontrolador con un microprocesador ATmega 2560 CMOS de 8 bits basado en la arquitectura RICS, entre sus características se encuantran:

- Voltaje de alimentación: 7 - 12 V
- Memoria flash: 256 KB
- Memoria EEPROM: 4 KB

- Memoria RAM: 8 KB

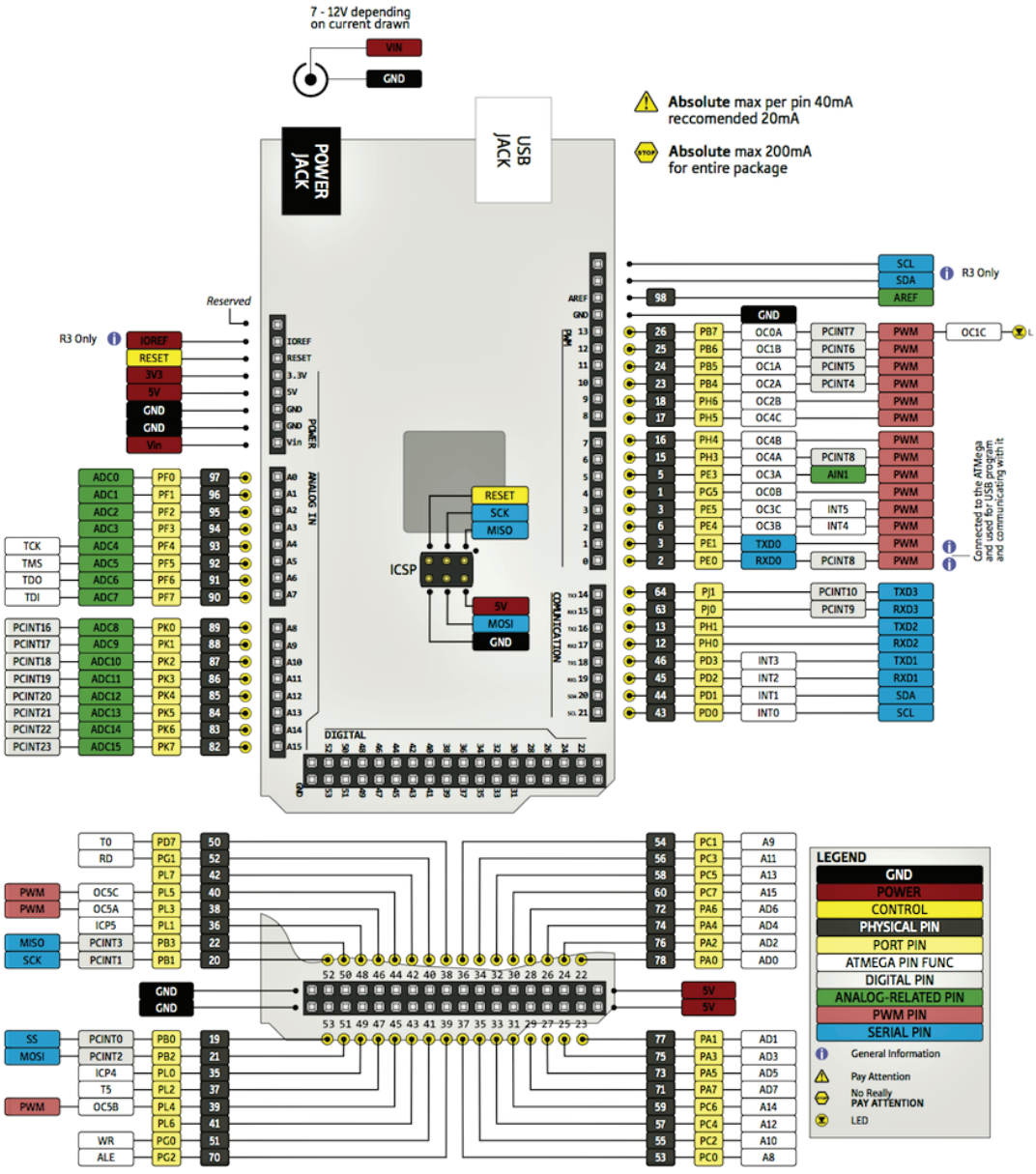


Figura C.7: Pinout Arduino Mega 2560

- Velocidad del cristal: 16 MHz
- Pines de proposito general: 86

- Canales PWM de 16 bits de resolución: 12
- Canales ADC: 16 de 10 bits
- Timers: 2 de 8 bits, 4 de 16 bits
- SPI: 5
- I2C: 1
- Serial USARTs: 4

No. Serial	0	1	2	3
Pin RX	0	19	17	15
Pin TX	1	18	16	14

Tabla C.1: Pines para serial Arduino Mega 2560

- Interrupciones externas: 6

No. Interrupción	0	1	2	3	4	5
Pin	2	3	21	20	19	18

Tabla C.2: Pines para interrupciones Arduino Mega 2560

C.5. Shield para el proyecto

El diseño de la PCB para el proyecto se realizó en el software Eagle. La Figura C.8 muestra el diseño esquemático. La Figura C.9 muestra el diseño de la PCB, tanto la vista superior (componentes), como la inferior (pistas) y también las dimensiones de esta.

El shield se realizó en una placa de fibra de vidrio.

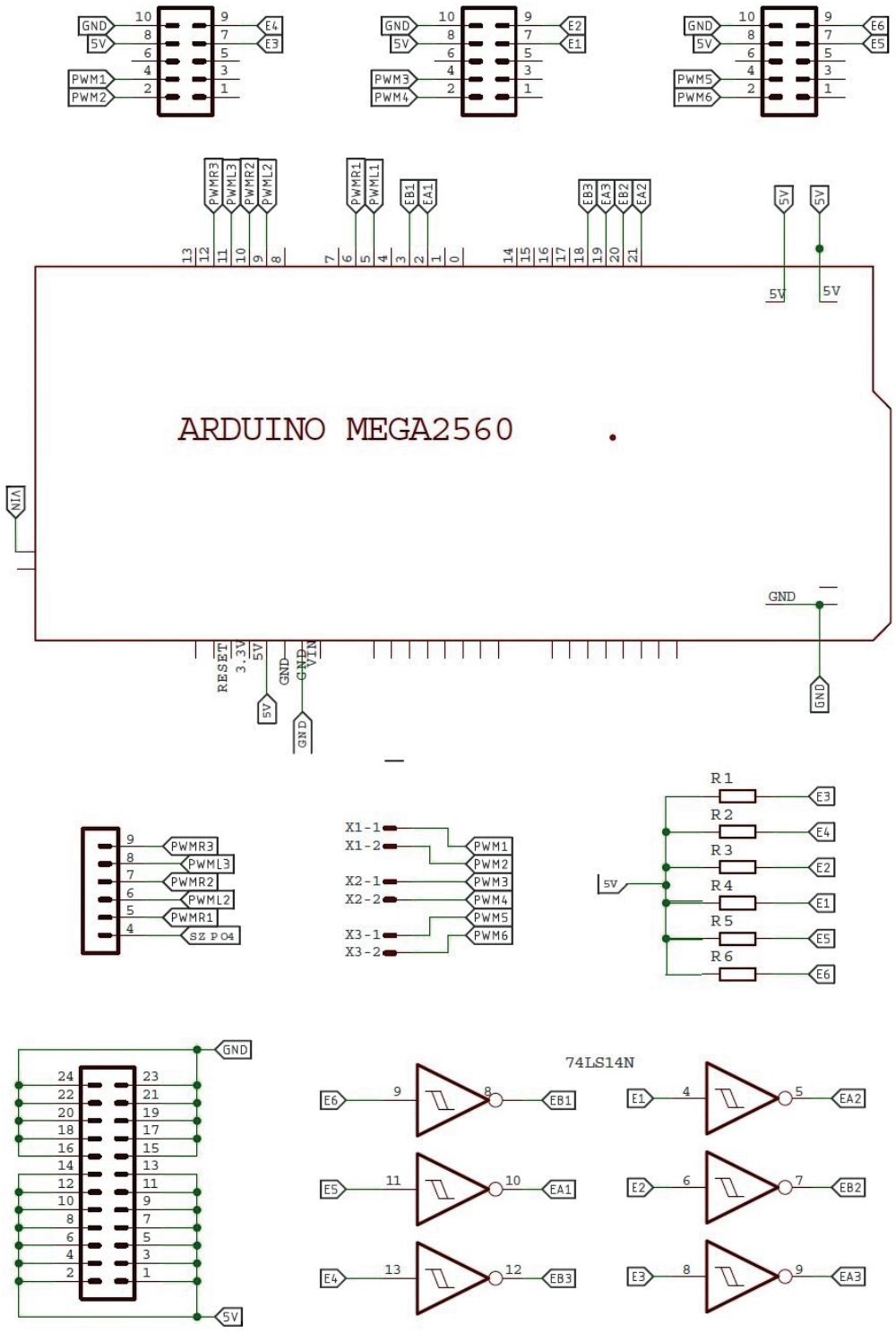


Figura C.8: Diseño de la PCB en Eagle

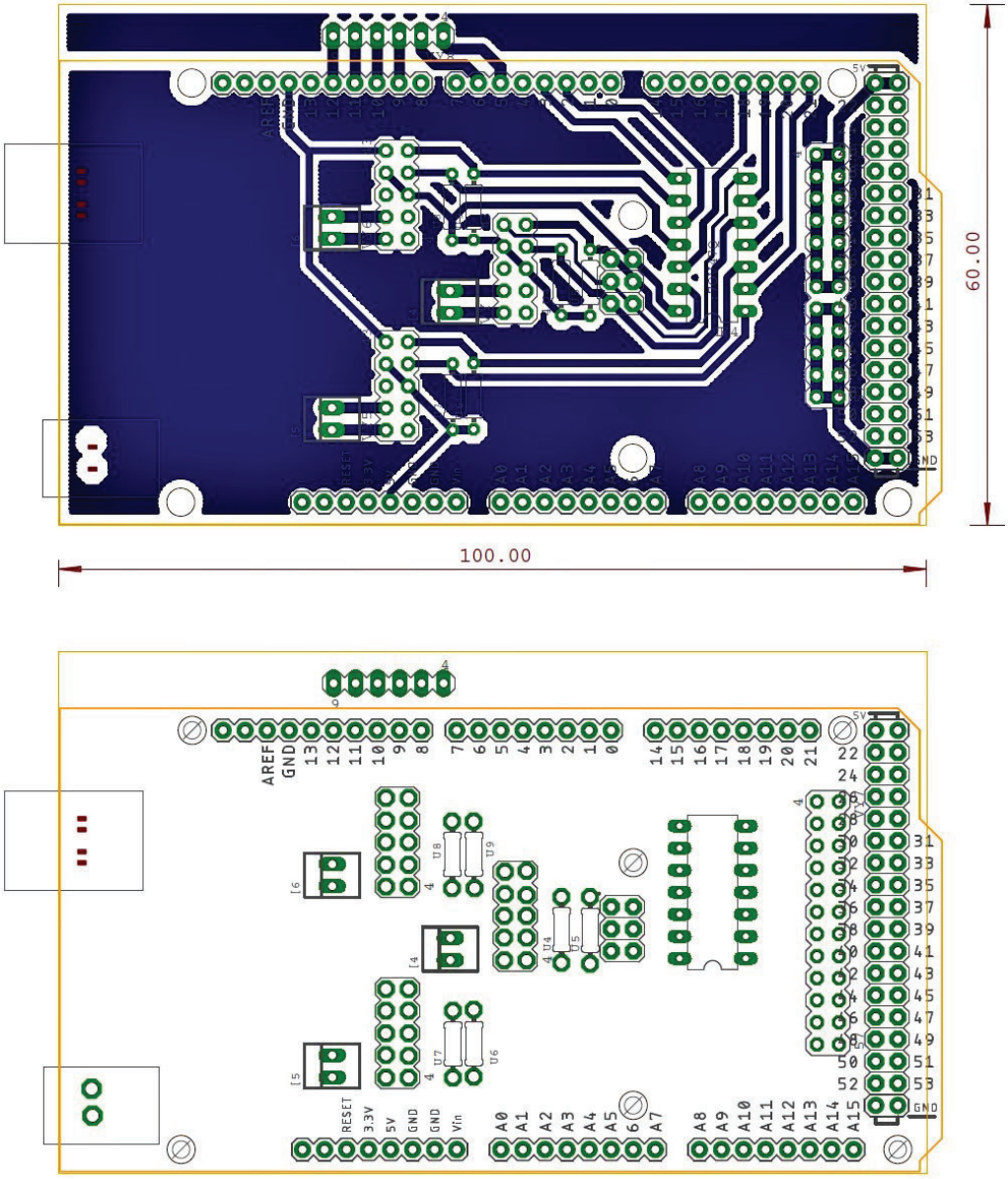


Figura C.9: Vista inferior y superior de la PCB

Apéndice D

Uso del software

D.1. MasterCAM

El uso del software MasterCAM 2019 se resume en los siguientes pasos:

Paso 1. Ejecutar el ícono de Master CAM 2019, se despliega la ventana como se ve en la Figura D.1. Tiene un entorno de fácil uso, similar a los softwares SolidWorks, AutoCAD, FreeCAD. Además, que los archivos de estos son compatibles.

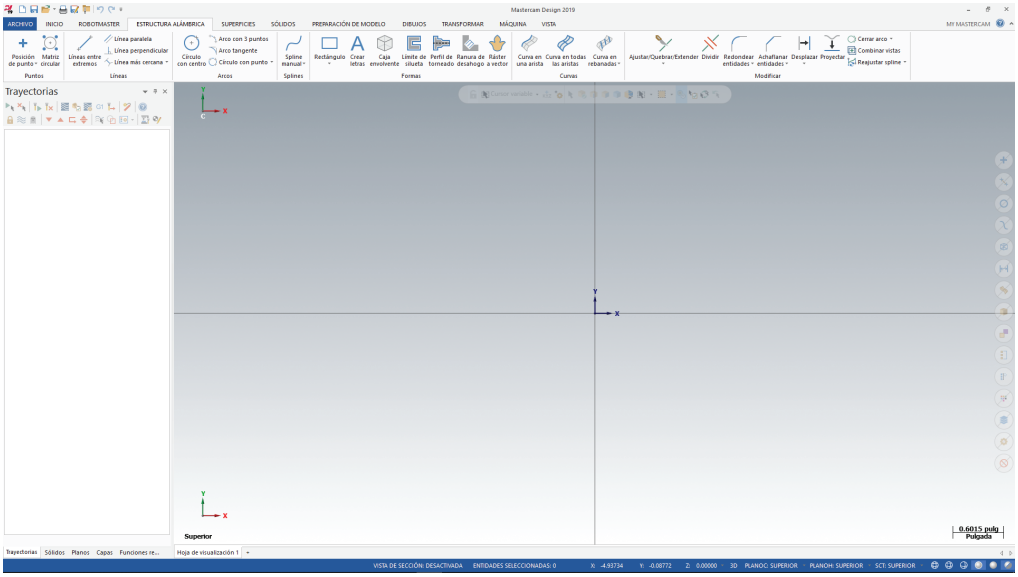


Figura D.1: Entorno MasterCAM

Paso 2. Una vez generada una pieza en 2D o 3D se visualiza como en la Figura D.2. Donde se ve el sistema de referencia en X, Y, Z. Con el scroll del mouse se puede rotar y aumentar o disminuir el tamaño de la pieza.

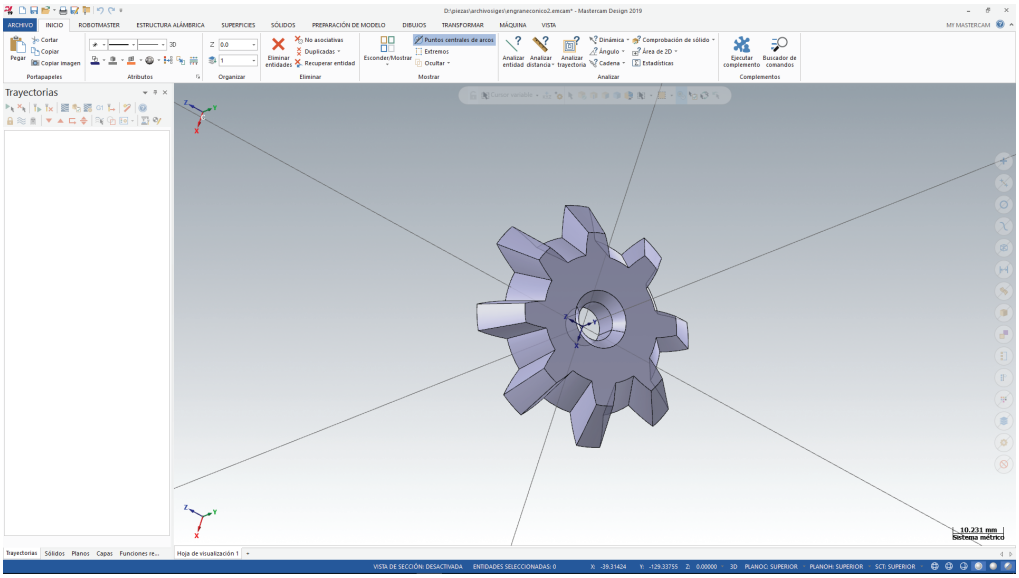


Figura D.2: Visualización pieza 3D en MasterCAM

Paso 3. Antes de seleccionar la máquina hay que modificar la referencia, el eje Z siempre va del lado donde se realizará el desbate. También se realizan las capas necesarias. Ahora se selecciona la máquina, la disponible por la licencia es el robot DENSO Figura D.3.

Paso 4. A continuación aparece una nueva ventana en la cual se podrá seleccionar el tipo de corte en 2D , 3D o multiejes Figura D.4, como se observa en la Figura D.5.

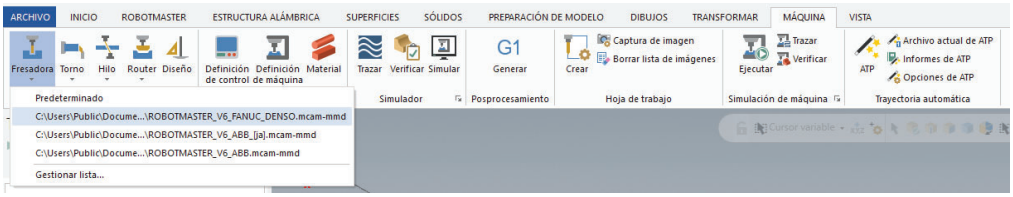


Figura D.3: Selección de máquina en MasterCAM

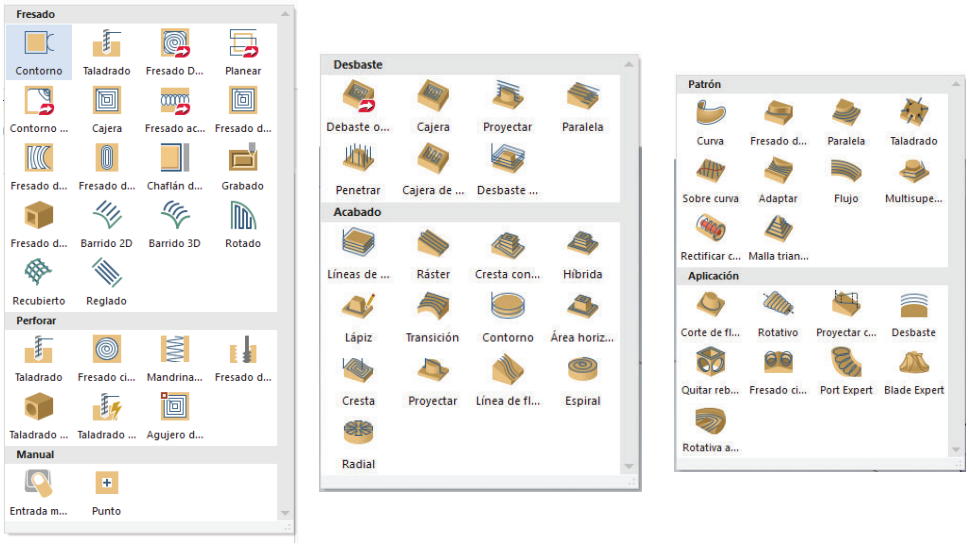


Figura D.4: Maquinados en 2D, 3D y multi ejes en MasterCAM

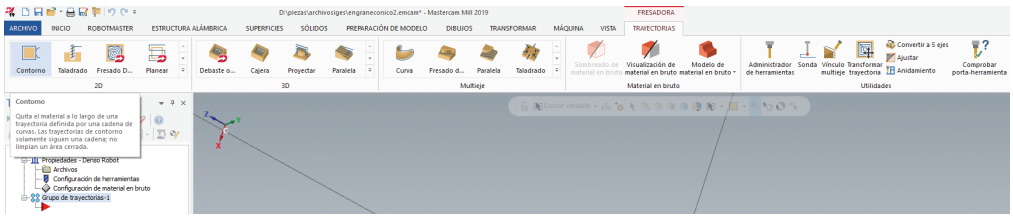


Figura D.5: Tipos de corte en MasterCAM

Paso 5. Se hace un corte de contorno, con ello se define el material bruto de la pieza de trabajo, se puede ingresar los límites o se puede hacer una caja envolvente la cual en automático define los límites del material bruto, seleccionando la pieza 3D como se ve en las siguientes Figuras D.6, D.7.

Paso 6. Lo siguiente es seleccionar el tipo de herramienta de corte, el número de herramienta, así como el tipo de material con el que se va a trabajar Figura D.8. Mediante los filtros se pueden seleccionar una o más herramientas, el software tiene definidas algunas, sin embargo, se pueden modificar estas.

En el caso de querer modificar los valores de la herramienta, se siguen los siguientes pasos.

Paso 7. En la opción editar herramienta se despliega una ventana como en la

Figura D.9, se deben modificar los parámetros. Se selecciona el botón siguiente y aparece una nueva ventana, aquí se modifican los diferentes tipos de velocidades.

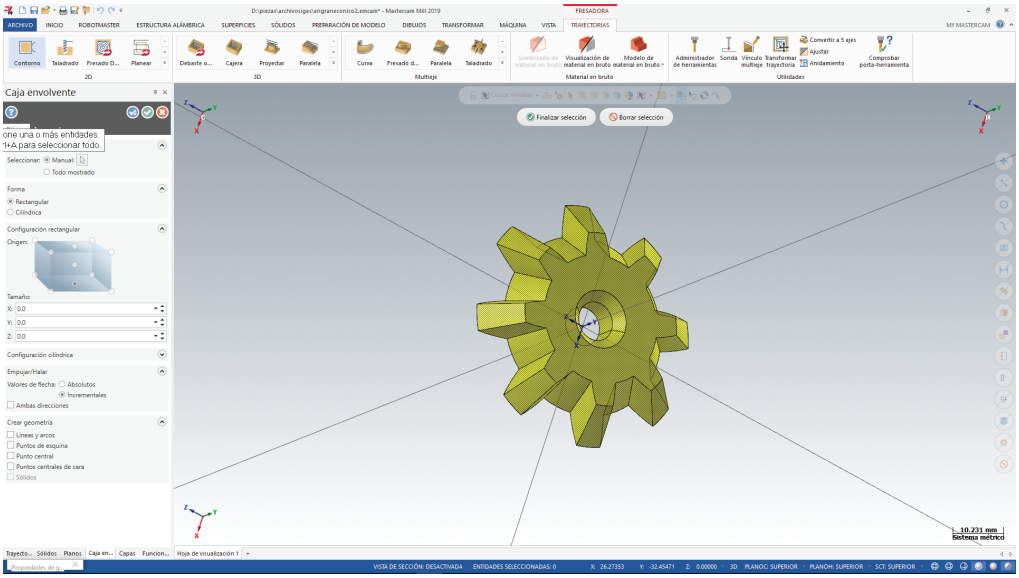


Figura D.6: Caja envolvente en MasterCAM

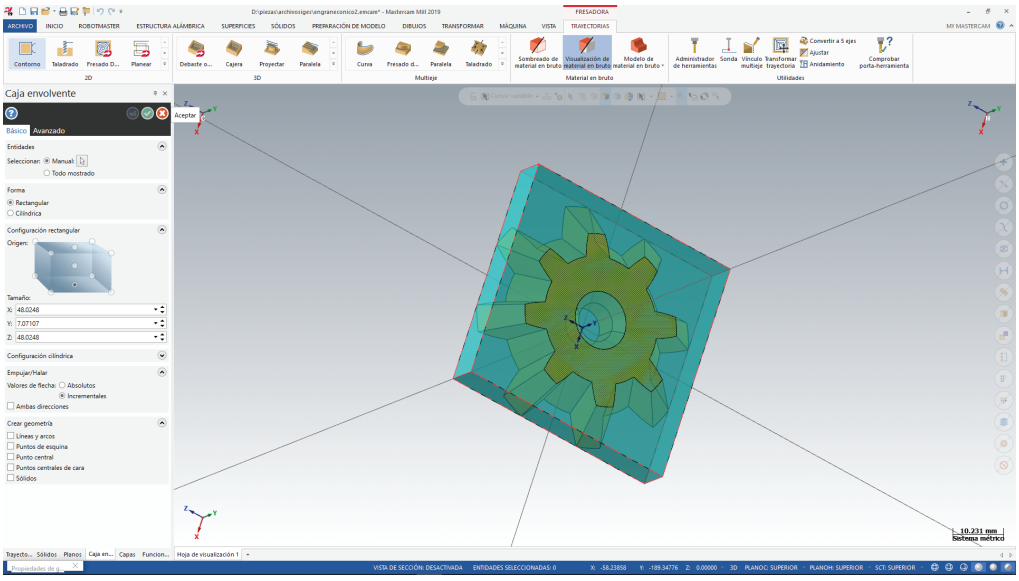


Figura D.7: Material bruto de la pieza de trabajo en MasterCAM

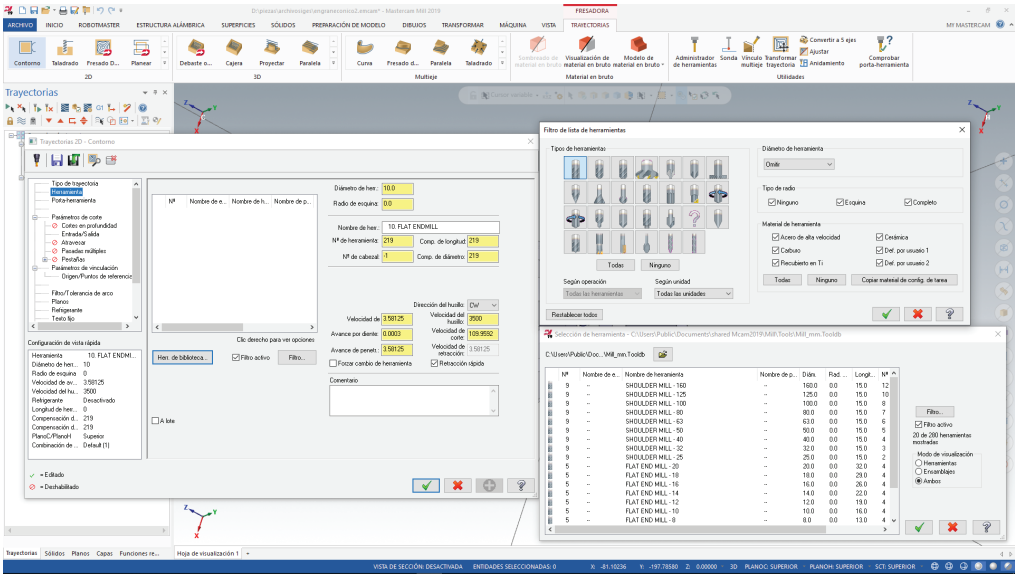
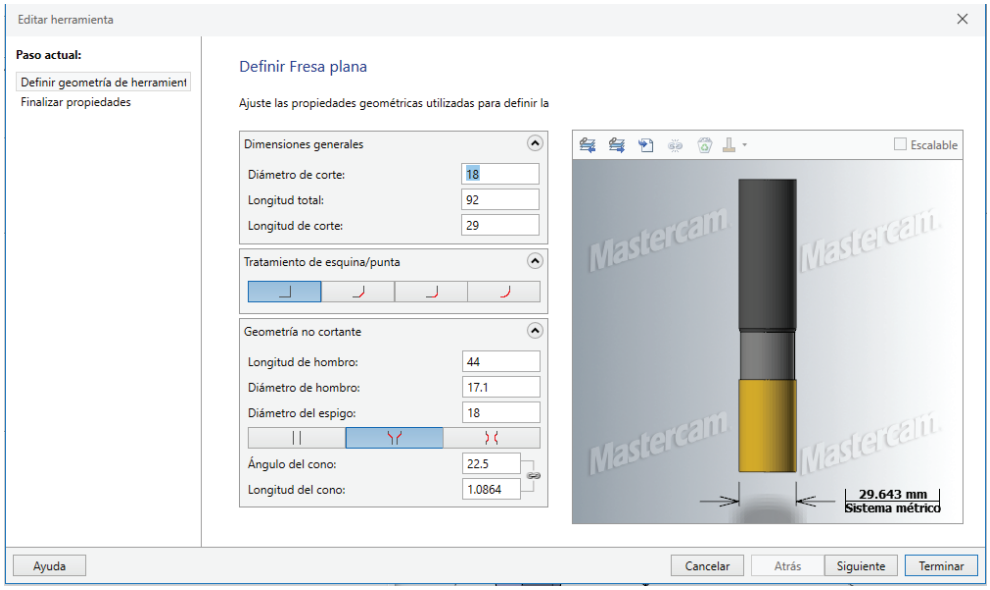


Figura D.8: Tipo de herramienta de corte en MasterCAM

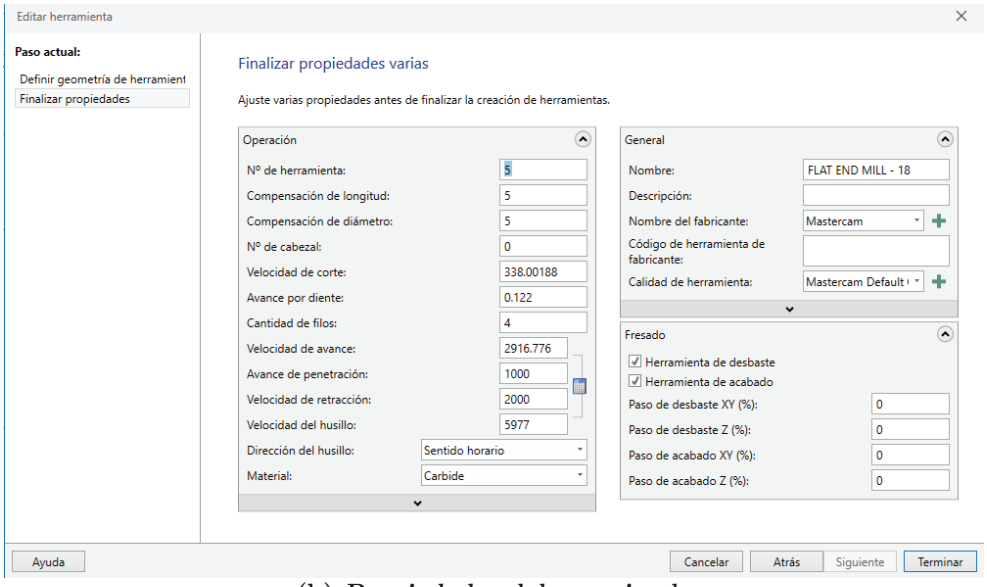
Paso 8. Lo siguiente es modificar el porta herramienta. MasterCAM incluye algunos porta-herramientas, en caso de que sea diferente el que uno trabaje se crea uno nuevo. En la pestaña nuevo porta-herramienta, se despliega una ventana como en la Figura D.10. Para el caso del robot académico DENSO se elimina el porta-herramienta ya que en el archivo CAD del efector final en RobotMaster ya cuenta con él.

Paso 9. Una vez realizadas estas modificaciones se procede a realizar un corte en 3D ya que esto ayuda a realizar cortes más rápidos para después realizar el corte multi-ejes. A continuación se definen los parámetros del corte 3D Figura D.11, así como los parámetros de entrada y salida Figura D.12. Una vez finalizada esta operación en la pantalla aparecen los cortes que se realizan a la pieza, así como los puntos de entrada y salida Figura D.13.

Paso 10. A continuación se realiza el corte en multi-ejes, es decir con los 6 DOF del robot. En este tipo de corte el efector final tiene la posibilidad de orientarse. Se definen de igual manera los parámetros de tipo de trayectoria FiguraD.14, patrón de corte Figura D.15, control del eje de herramienta Figura D.16, control de colisiones y vinculación Figura D.17. Una vez finalizado, en la pieza aparecen los cortes en multi ejes que se realizarán Figura D.18.



(a) ***Propiedades de la herramienta***



(b) ***Propiedades del maquinado***

Figura D.9: Ajustes de maquinado en MasterCAM

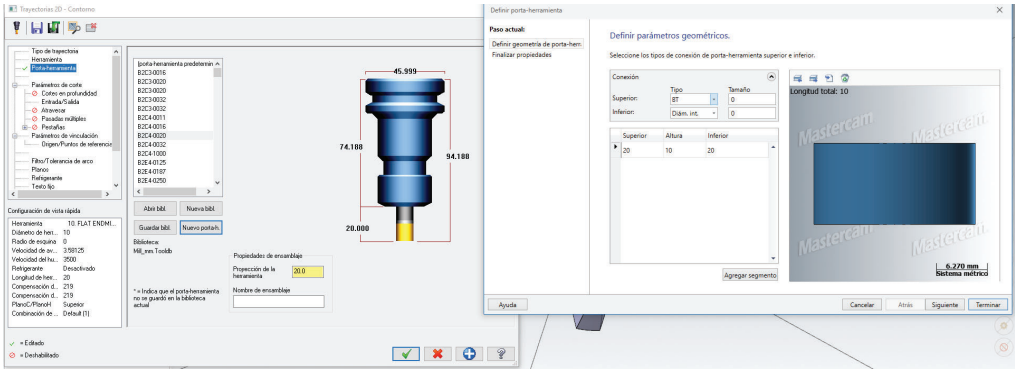


Figura D.10: Tipo de herramienta de corte en MasterCAM

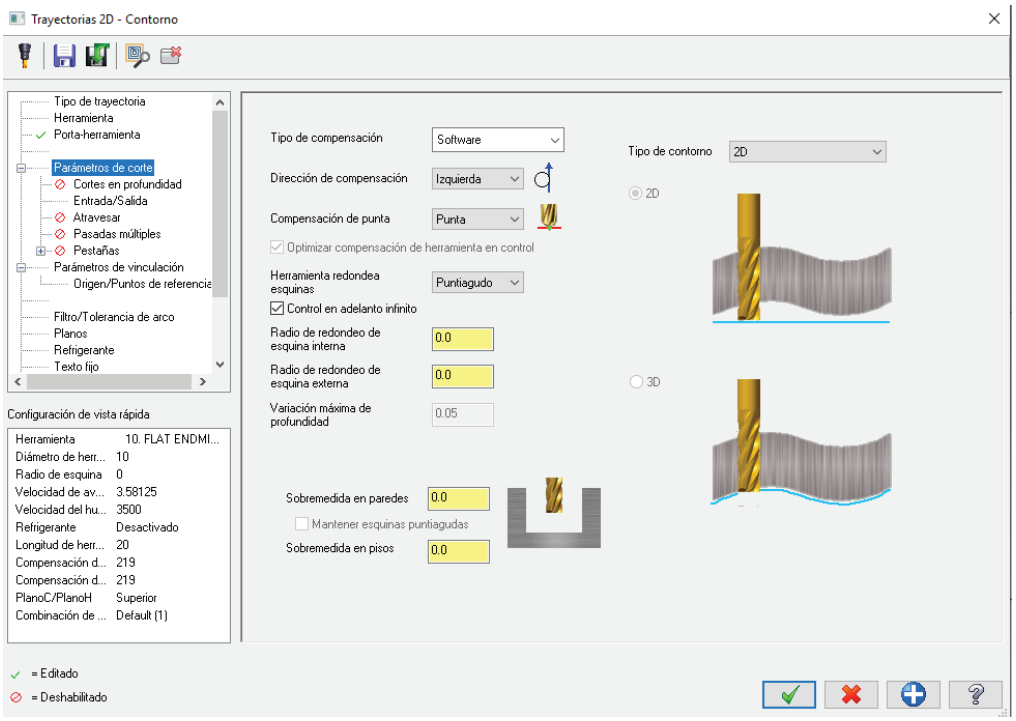


Figura D.11: Ajustes de porta herramienta en MasterCAM

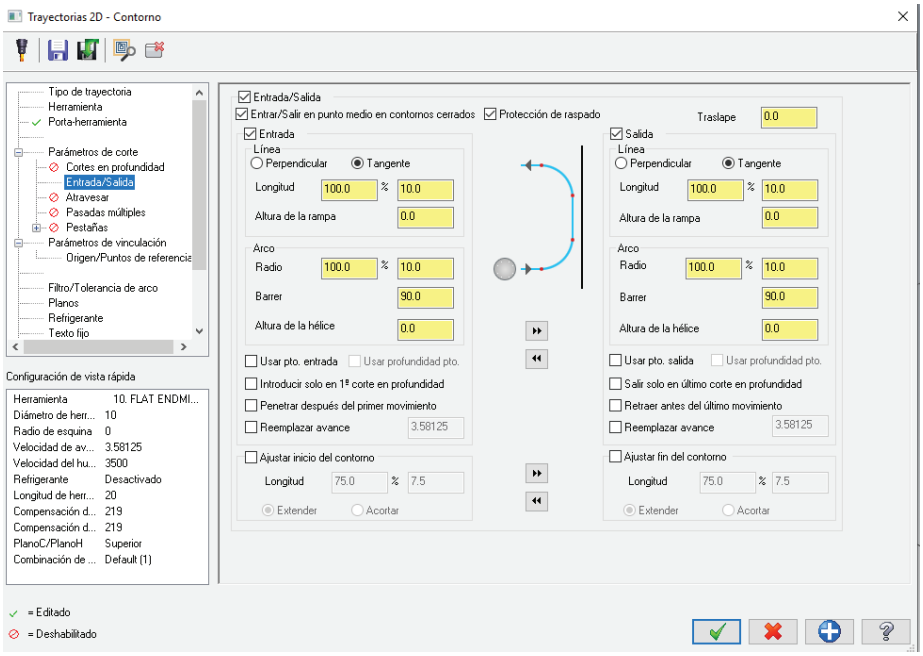


Figura D.12: Parámetros de corte en MasterCAM

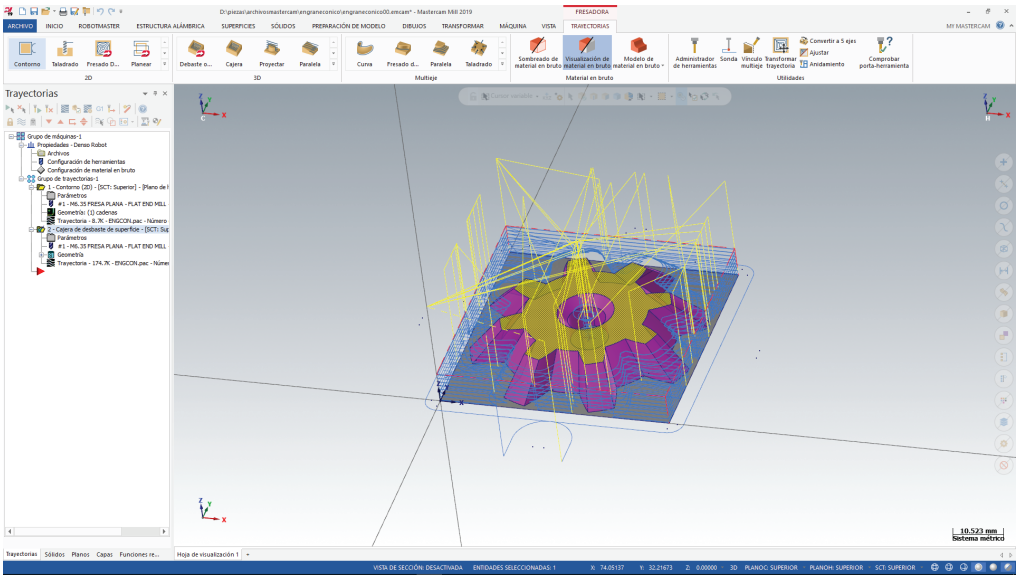


Figura D.13: Parámetros entrada/salida de corte en MasterCAM

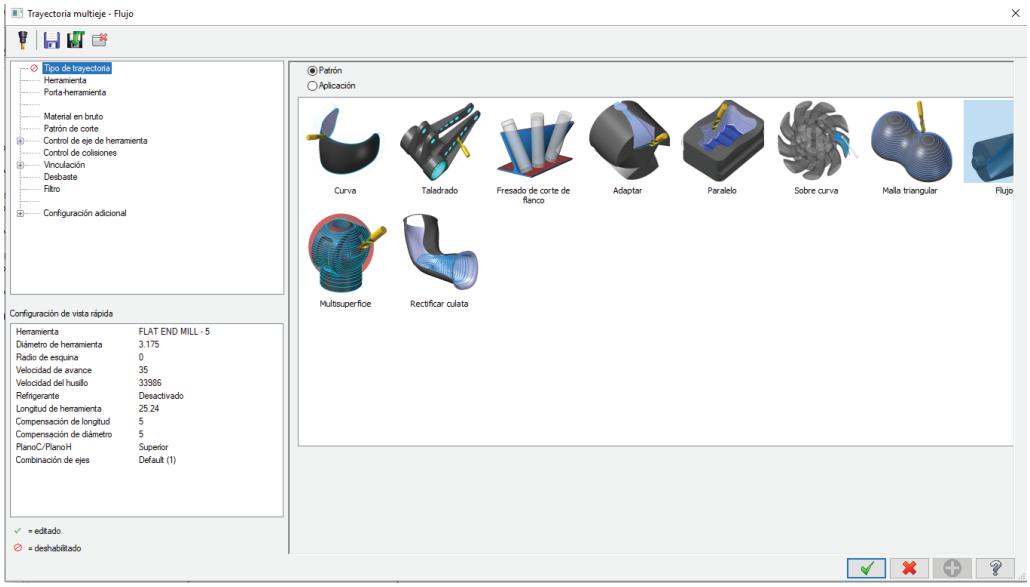


Figura D.14: Visualización del maquinado 3D en MasterCAM

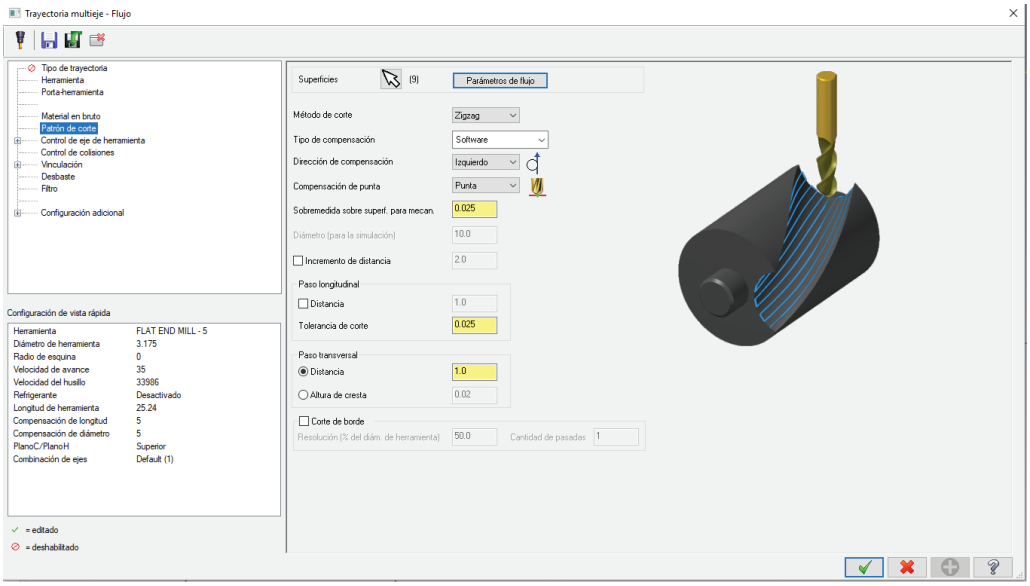


Figura D.15: Selección de trayectoria multi eje en MasterCAM

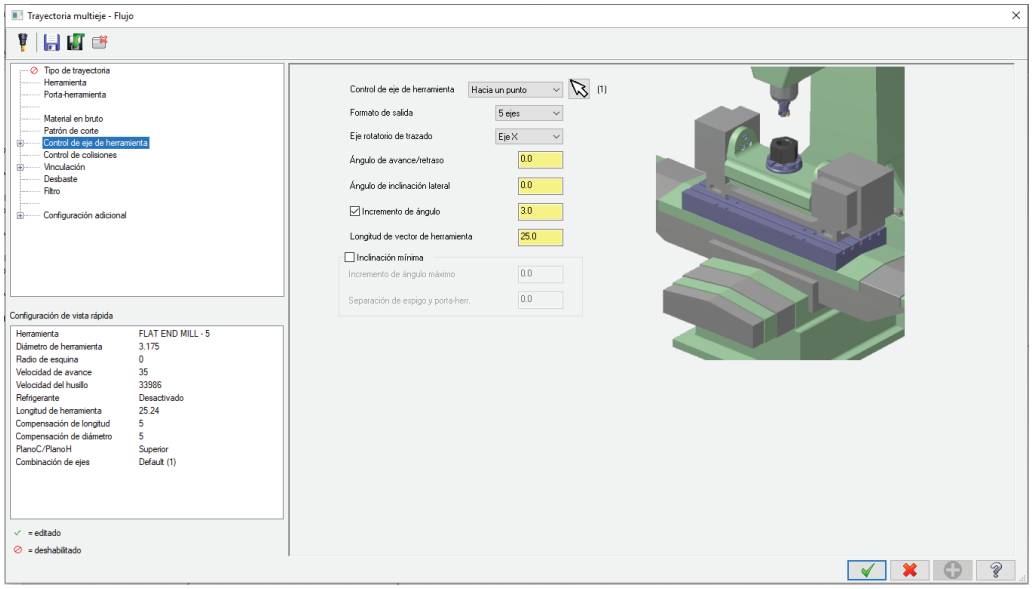


Figura D.16: Patrón de corte multi eje en MasterCAM

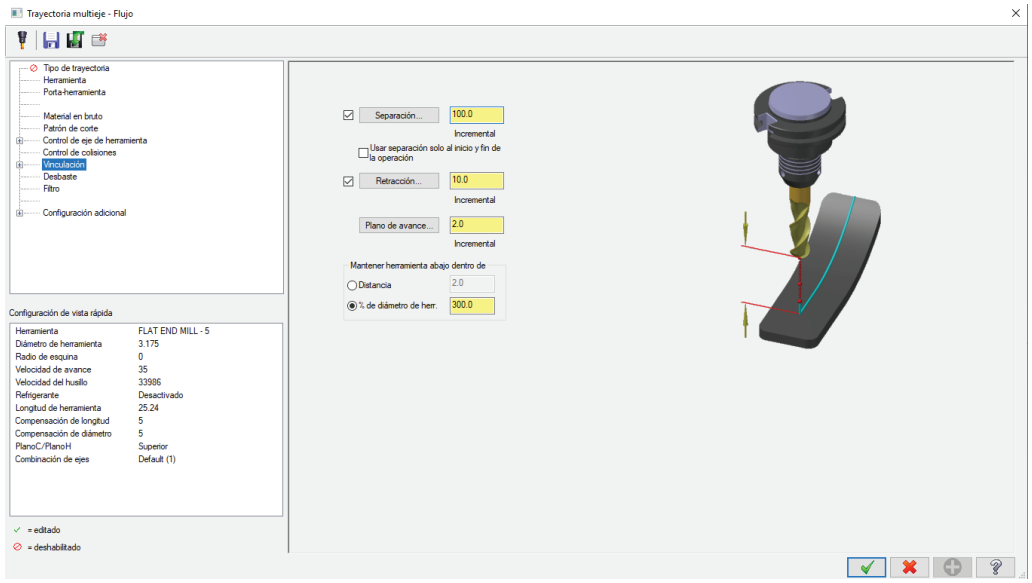


Figura D.17: Control del eje de herramienta de corte en MasterCAM

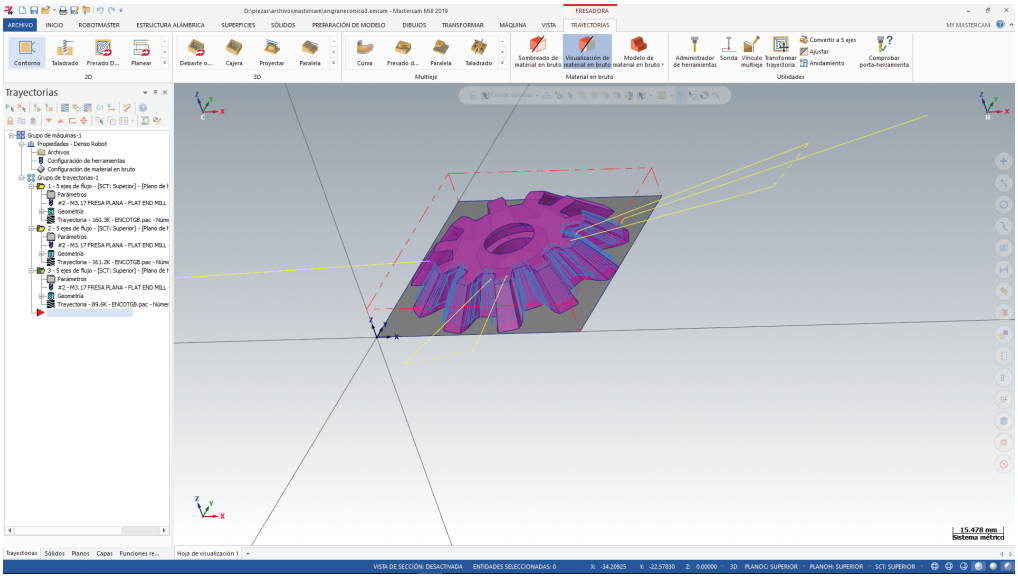


Figura D.18: Vinculación de corte en MasterCAM

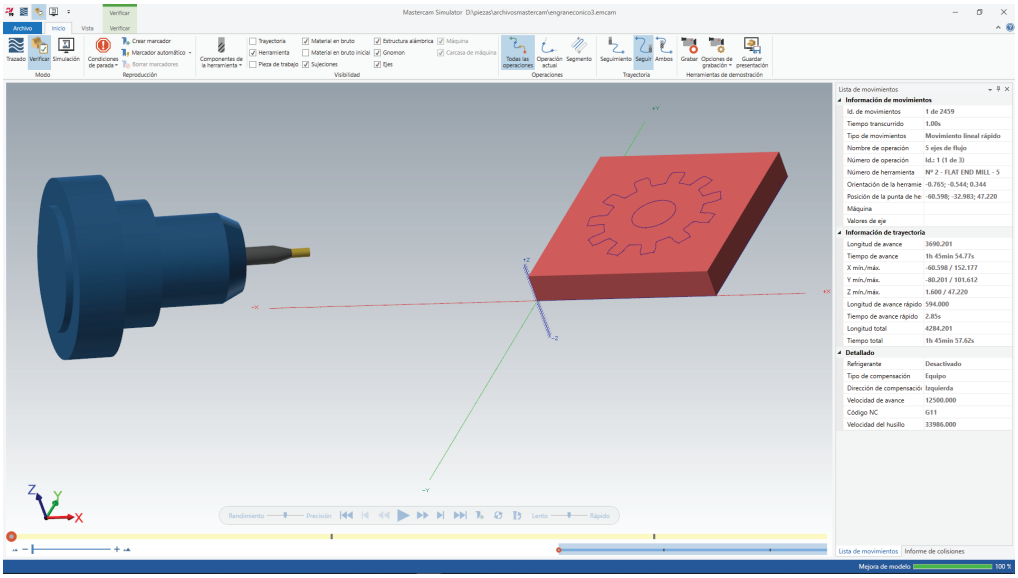


Figura D.19: Visualización del maquinado multi eje en MasterCAM

Paso 11. Ahora sigue ver la simulación del corte realizado. En la pestaña verificar las operaciones seleccionadas, se le da un click y aparece una ventana como en la Figura D.19, donde se puede observar toda la información detallada de movimientos, trayectorias, material bruto, la herramienta así como el porta

herramienta de corte. Para ver cómo quedará la pieza, se da play a la simulación y se ve como se realiza cada corte Figura D.20.

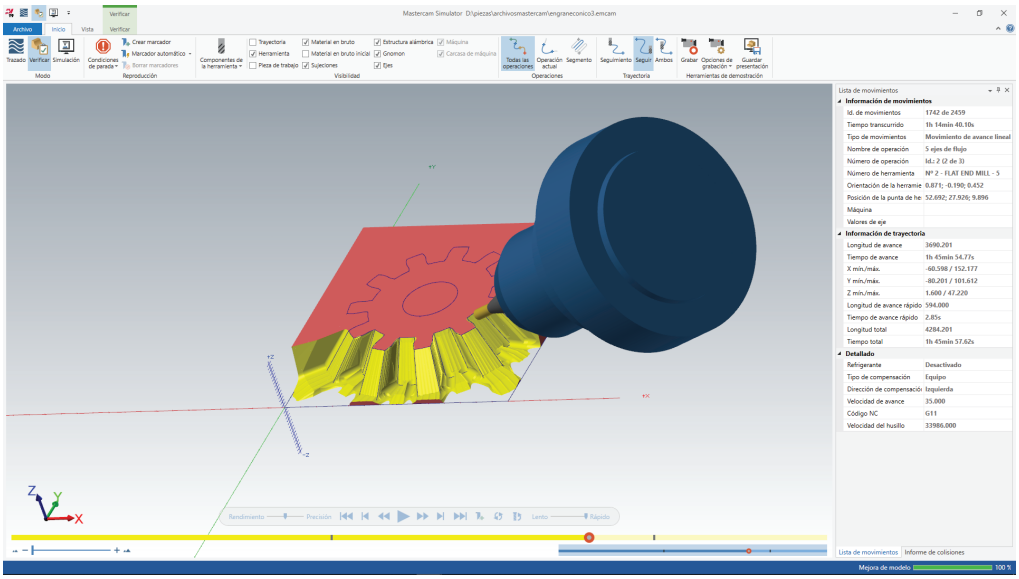


Figura D.20: Ventana de simulación en MasterCAM

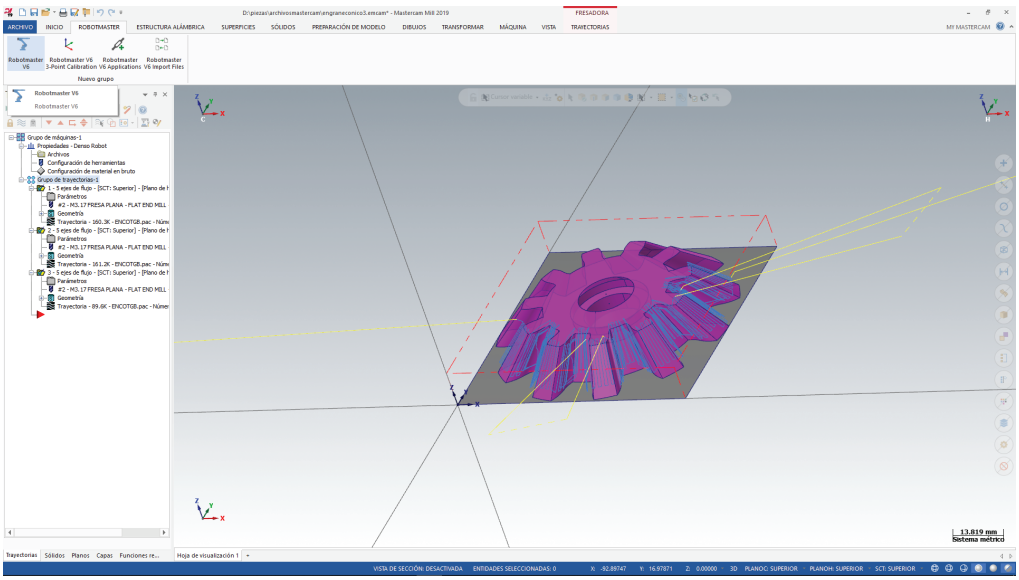


Figura D.21: Pestaña para cambiar de MasterCAM a RobotMaster

Paso 12. Sólo resta regresar al entorno de MasterCAM y seleccionar la pes-

taña que dice RobotMaster, de los íconos que despliega seleccionar RobotMaster V6 Figura D.21, con ello se abre directamente el software RobotMaster.

D.2. RobotMaster

Paso 1. Una vez abierto el software RobotMaster, se ve una ventana como en la Figura D.22, se procede a manipular los ajuste de simulación, donde se activa o desactiva el entorno de trabajo Figura D.23.

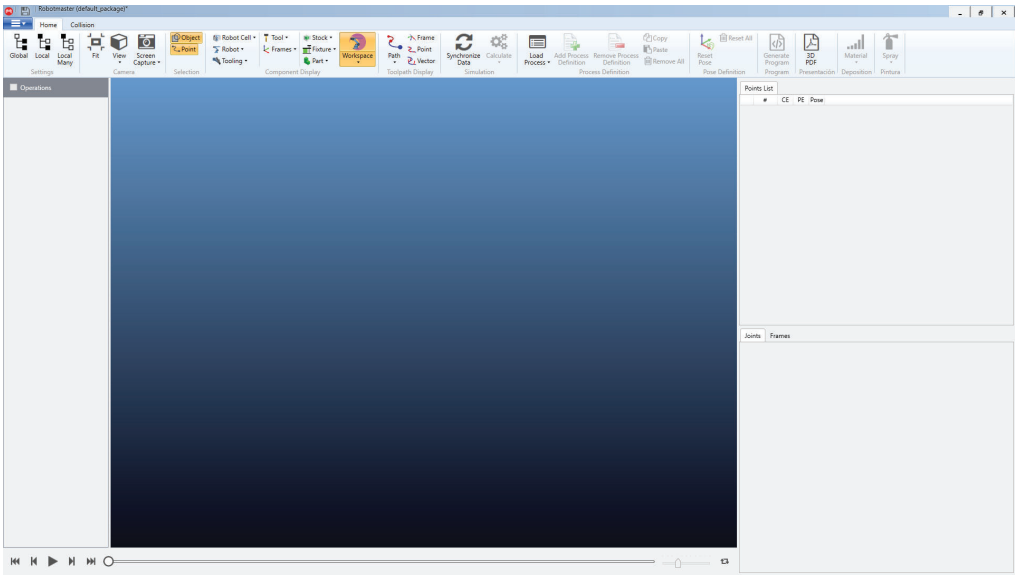


Figura D.22: Entorno RobotMaster

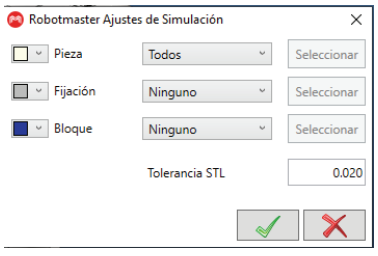


Figura D.23: Ajustes de simulación en RobotMaster

Paso 2. A continuación, de la pestaña Global, se selecciona y se despliega una ventana como en la Figura D.24, en esta ventana se selecciona tanto el robot a utilizar, así como el efector final, en este caso seleccionamos el robot DENSO

y el efector final BASEROBOT Figura D.25. En el apartado de Frame Data, se establece la distancia que existe entre la pieza de trabajo y la coordenada base del robot. También aquí se puede modificar la posición y orientación del efector final Figura D.26.

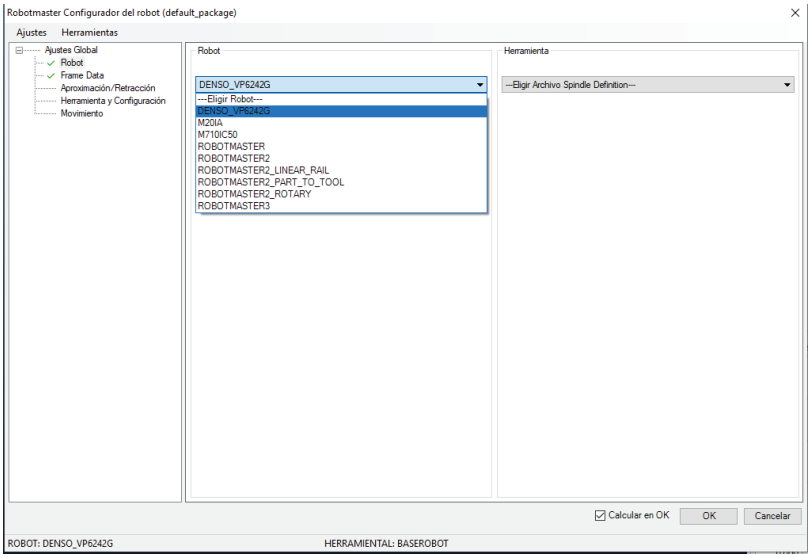


Figura D.24: Selección del robot en RobotMaster

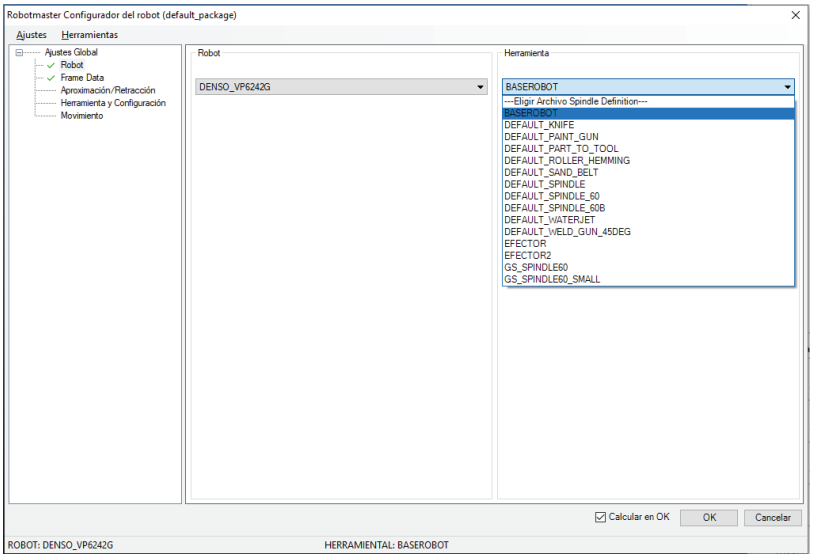


Figura D.25: Selección del efector final en RobotMaster

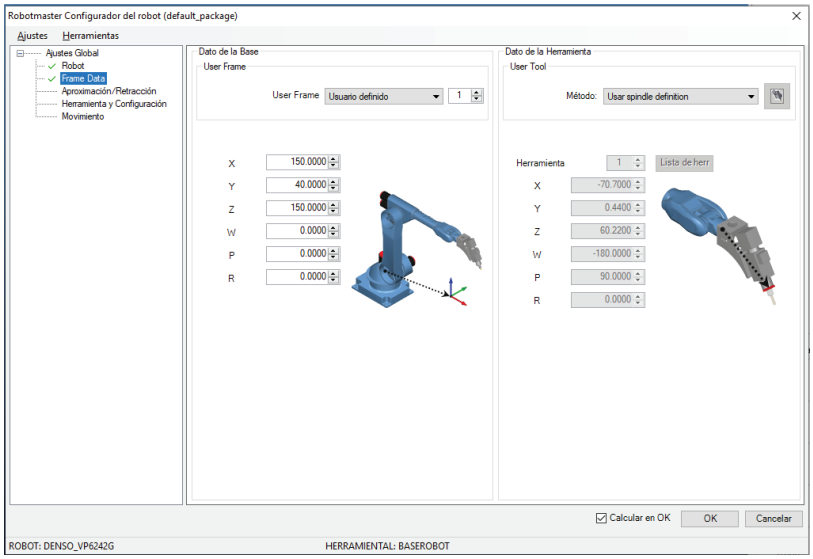


Figura D.26: Configuración de la pieza de trabajo en RobotMaster

Paso 3. Ahora en la pestaña colisión, se activa los diferentes parámetros, esto con el fin de que no existan choques entre el efector final tanto con la pieza de trabajo así como el espacio de trabajo Figura D.27.

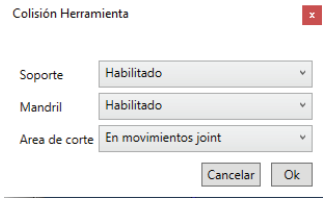


Figura D.27: Ajuste de colisiones en RobotMaster

Los pasos anteriores sólo fueron para activar y configurar los parámetros del robot, concluido esto debe de aparecer en la ventana de trabajo tanto el robot completo (con efector final) como la pieza de trabajo como se ve en la Figura D.28. Pero aún falta realizar las operaciones de corte, como se ve en la imagen operación 1 y 2 aparecen con un signo de interrogación.

Paso 4. En la pestaña local, se selecciona y aparece una ventana como en la Figura D.29, en el apartado de optimización se activan las opciones de mapas, esto para no haya problemas cuando se generé el programa del robot. Se pueden modificar más parámetros, una vez finalizado se selecciona en calcular. Finalizado

los cálculos aparece en la ventana regiones de diferentes colores, donde cada color está vinculado con los mapas seleccionados anteriormente y aparece una línea la cual tiene la finalidad de mover la orientación del efector final para que este no tenga problemas para realizar el corte. Entonces esta línea debe de estar en el espacio en blanco como se ve en la Figura D.29. Se acepta en la pestaña ok.

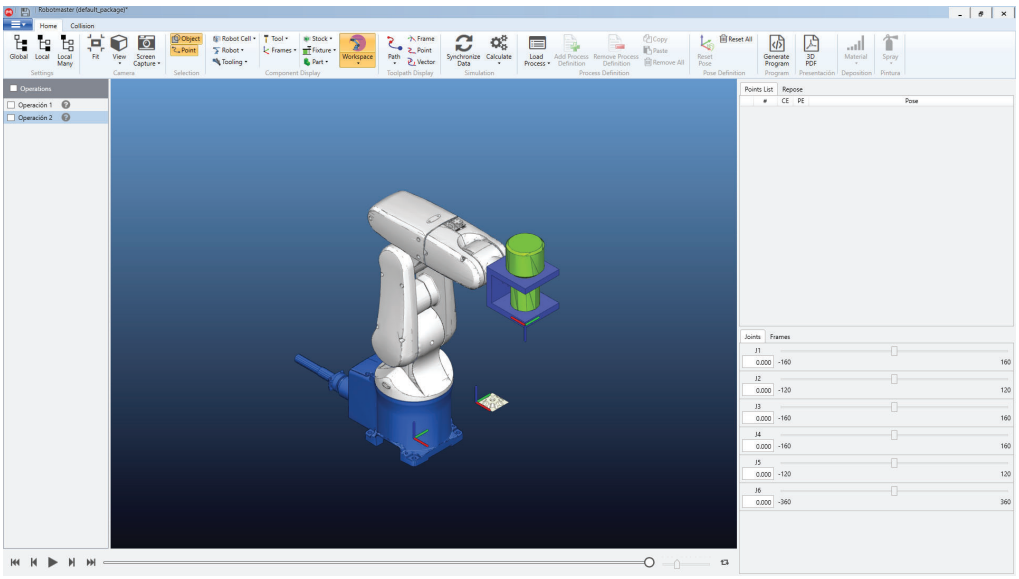


Figura D.28: Robot, efector final y pieza de trabajo en RobotMaster

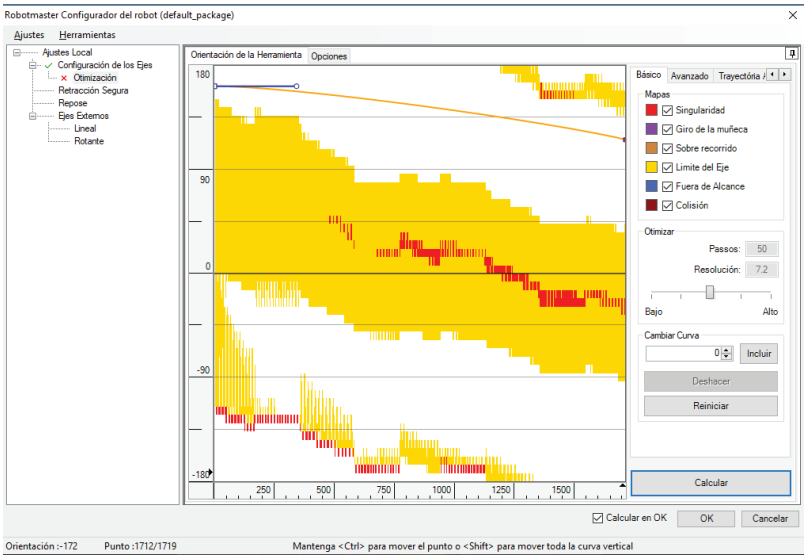


Figura D.29: Optimización de trayectorias en RobotMaster

El paso 4 se repite para las diferentes operaciones de corte, una vez termidas todas aparece en la ventana de trabajo las diferentes trayectorias de corte Figura D.30.

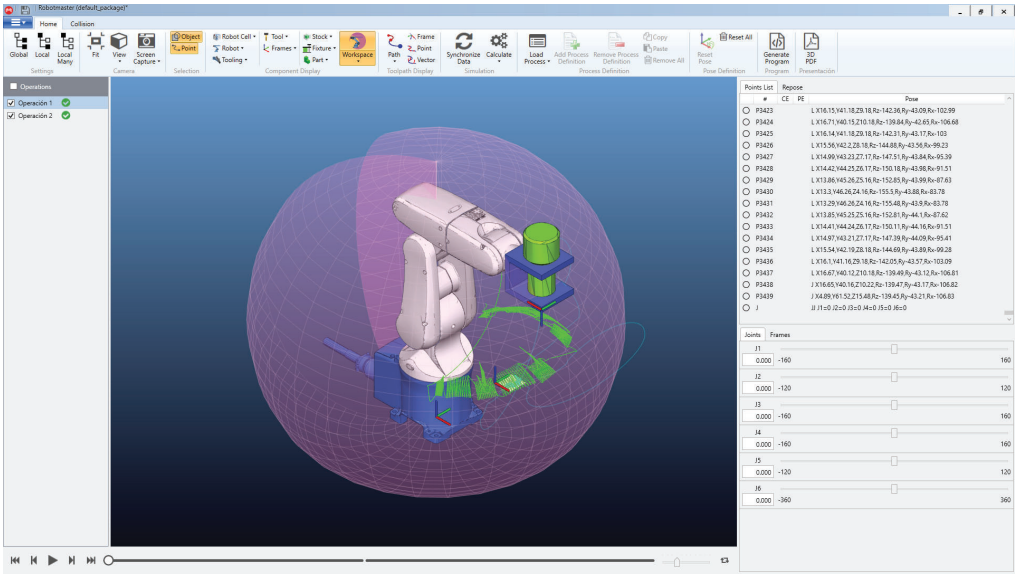


Figura D.30: Visualización de trayectorias de corte en RobotMaster

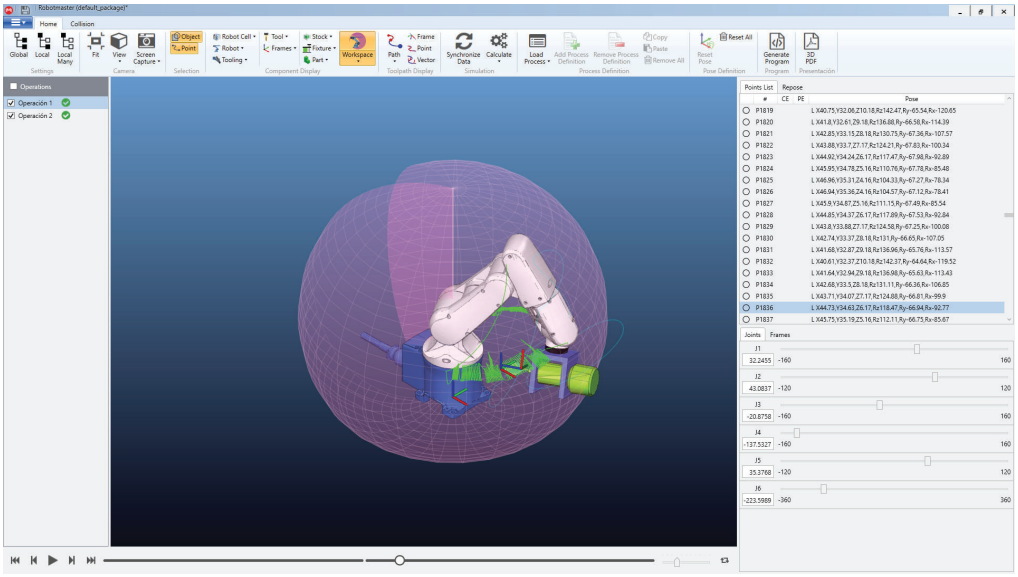


Figura D.31: Simulación de las trayectorias de corte en RobotMaster

Ahora del lado de operaciones ya no aparece el signo de interrogación, se cambia por una paloma en color verde. En el caso de que existan problemas con la realización del corte aparecería un tache en color rojo. A la derecha de la ventana aparece el total de posiciones y trayectorias que realizará el robot.

Paso 5. Lo siguiente es ver la parte de simulación para verificar que no haya problemas con el robot, la pieza de trabajo, el área de trabajo y el efector final Figura D.31.

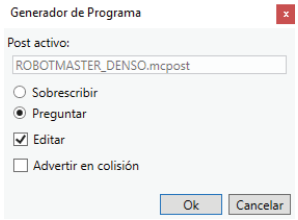


Figura D.32: Generación de programa de robot en RobotMaster

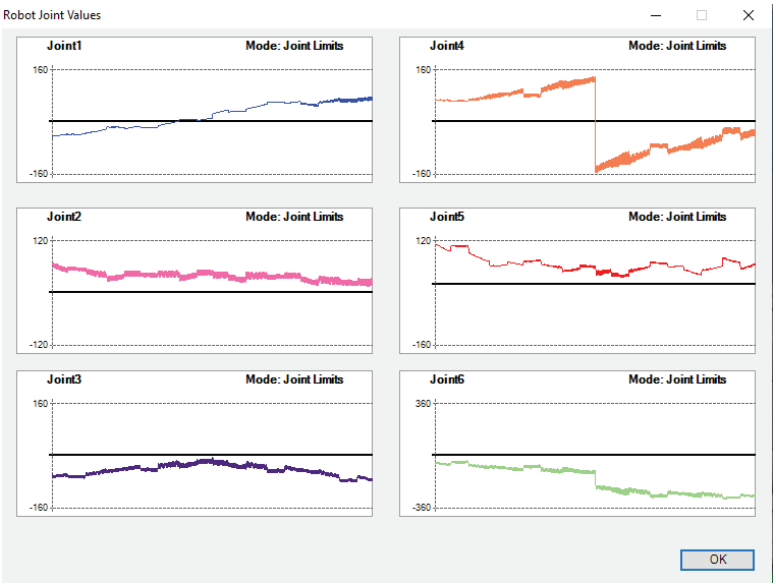


Figura D.33: Movimiento de cada articulación en RobotMaster

Paso 6. Por último queda generar el programa de robot, se selecciona la pestaña generar programa y aparece la ventana como en la Figura D.32, donde nos pregunta si queremos un programa nuevo o modificar algún programa anterior. Una vez aceptado aparece la ventana Figura D.33 que muestra los ángulos de giro

de cada articulación del robot para realizar la operación.

En MasterCAM aparece la siguiente Figura D.34 que nos pregunta si queremos que se generen subprogramas debido al tamaño del programa, lo que esto hará es dividir el programa en varias partes para que el controlador no tenga problemas cuando lea el programa. Una vez aceptado se abre el programa Code Expert, aquí muestra como quedo el divido el programa y en la carpeta donde se guardaron los archivos aparecen estas partes de programa Figura D.35.

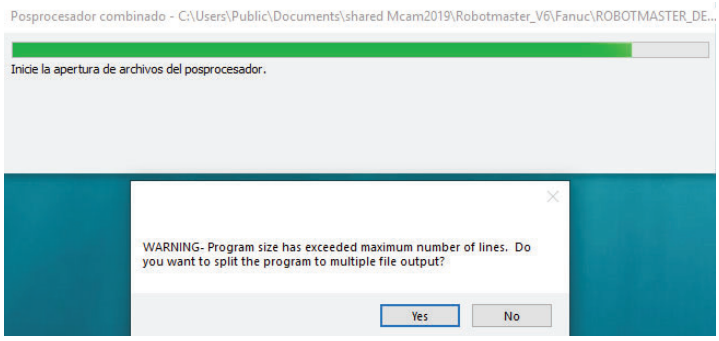


Figura D.34: Postprocesador en MasterCAM

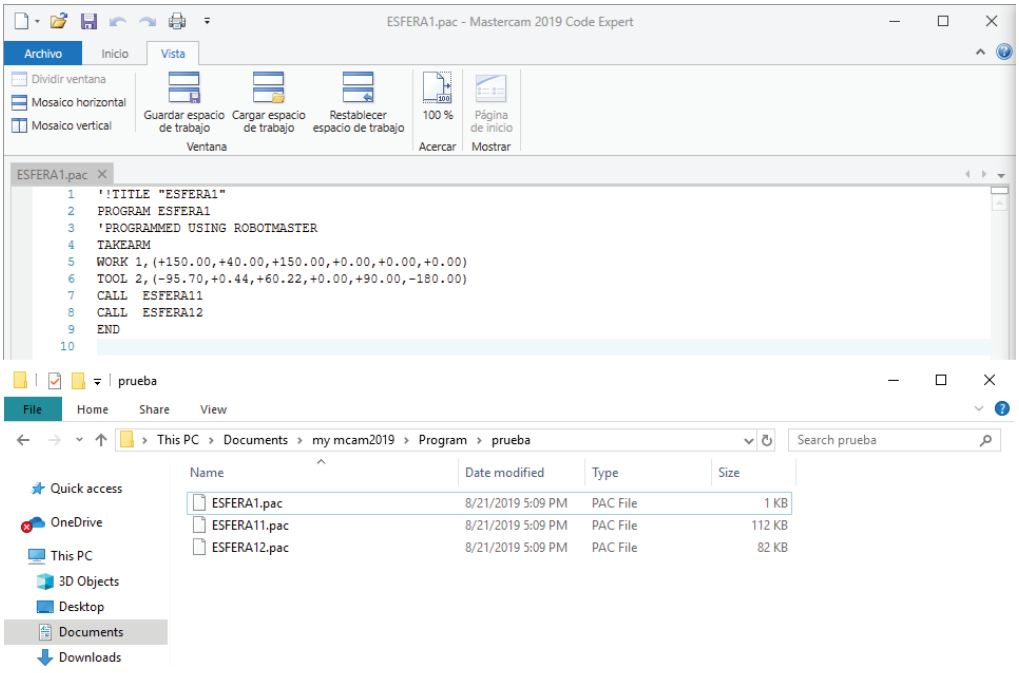


Figura D.35: Archivos generados en Code Expert

D.3. WinCaps iii

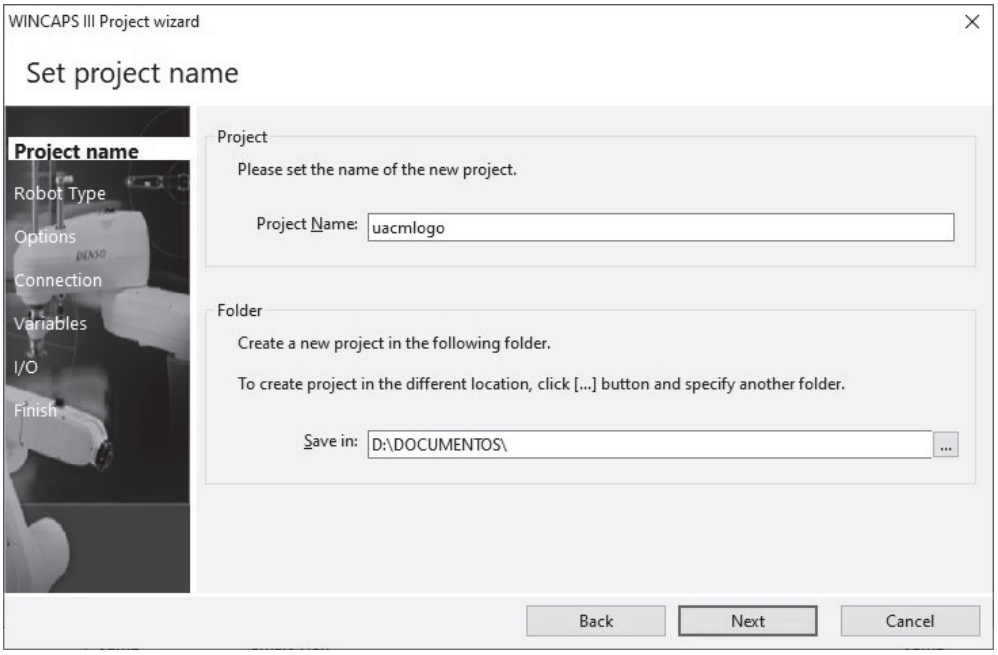


Figura D.36: Creación de un nuevo proyecto en Wincaps iii y dirección del archivo

Paso 1. En el software WInCaps iii, lo primero es realizar un nuevo proyecto y aparecerá una ventana como en la Figura D.36, donde se inserta un nombre al proyecto y se selecciona la dirección de la carpeta donde quedarán los archivos. El nombre del proyecto sólo puede contener letras tanto mayúsculas como minúsculas y no puede contener algún otro caracter.

Paso 2. Lo siguiente es seleccionar el modelo del brazo robótico, este va estar contenido dentro de la carpeta del modelo del controlador, en este caso el modelo del controlador es RC7M y del robot es VP-6442 Figura D.37.

Paso 3. A continuación sigue seleccionar las especificaciones del controlador, así como si existen DOF extras. También se selecciona la versión del compilador Figura D.38.

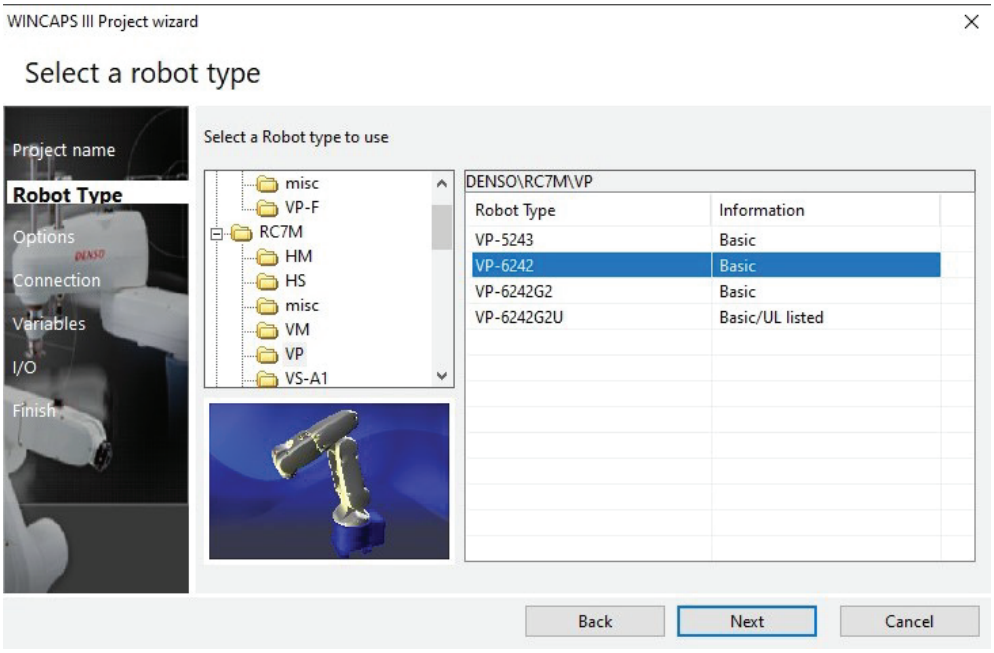


Figura D.37: Selección del tipo de robot en Wincaps iii

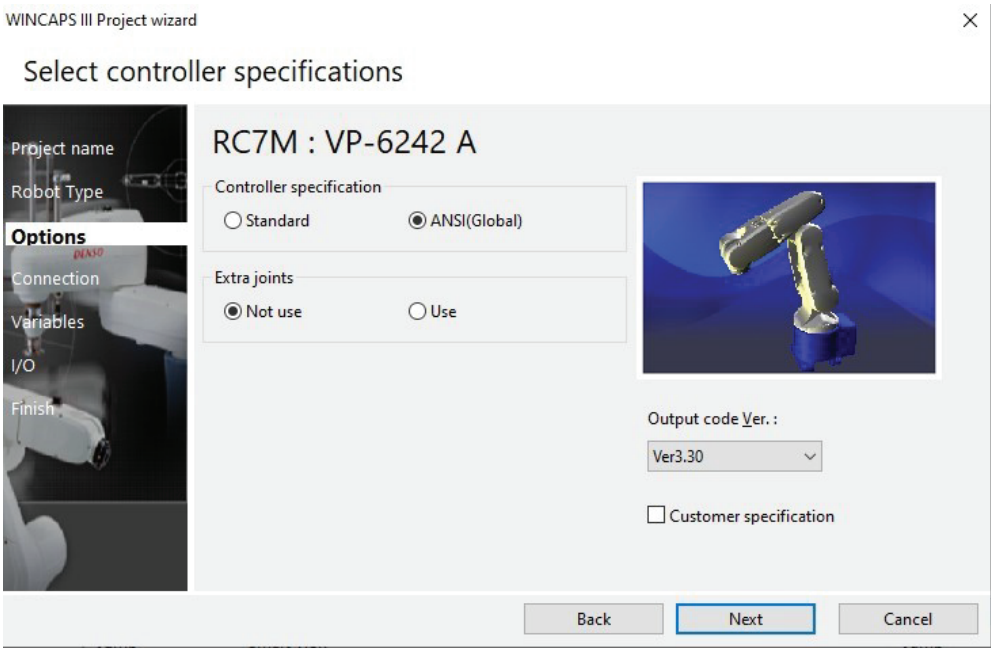


Figura D.38: Selección y especificaciones del robot en Wincaps iii

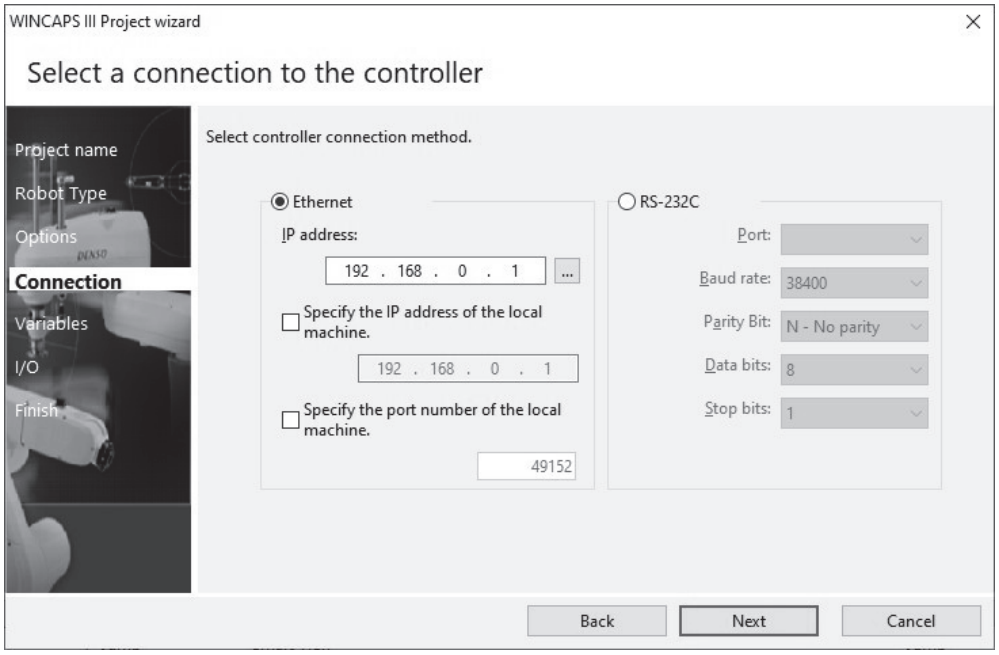


Figura D.39: Tipo de conexión y dirección en Wincaps iii

Paso 4. Ahora se selecciona el tipo de comunicación entre el controlador y la computadora y se especifica la dirección IP del controlador Figura D.39. Para saber los cambios a realizar tanto en la computadora como en el controlador, consultar [DR0 (4)].

Paso 4. La última ventana que aparece muestra el contenido del proyecto Figura D.40. Se finaliza y aparece una ventana como en la Figura D.41. En esta ventana se pueden hacer, agregar o editar programas para el robot. En el centro de la ventana general se observan las ventanas de cada uno de los programas generados, así como el robot a utilizarse. En la parte inferior del lado izquierdo se ven los archivos disponibles, en el centro los parámetros para el espacio de trabajo y la herramienta del efector final, del lado derecho aparece las entradas o salidas. También se pueden agregar otra variables, ya sea errores, posiciones, entre otros, consultar [DR0 (4)].

Teniendo ya los archivos y las modificaciones del espacio de trabajo y efector final, sigue cargar los archivos al controlador. En la pestaña "tool"se activa la comunicación en "enable online-debug mode", ahora se tranfieren los archivo en la pestaña "connect". Una vez realizado esto en el teach pendant se cambia al modo automático. Ahora se puede desde la computadora activar o desactivar al robot.

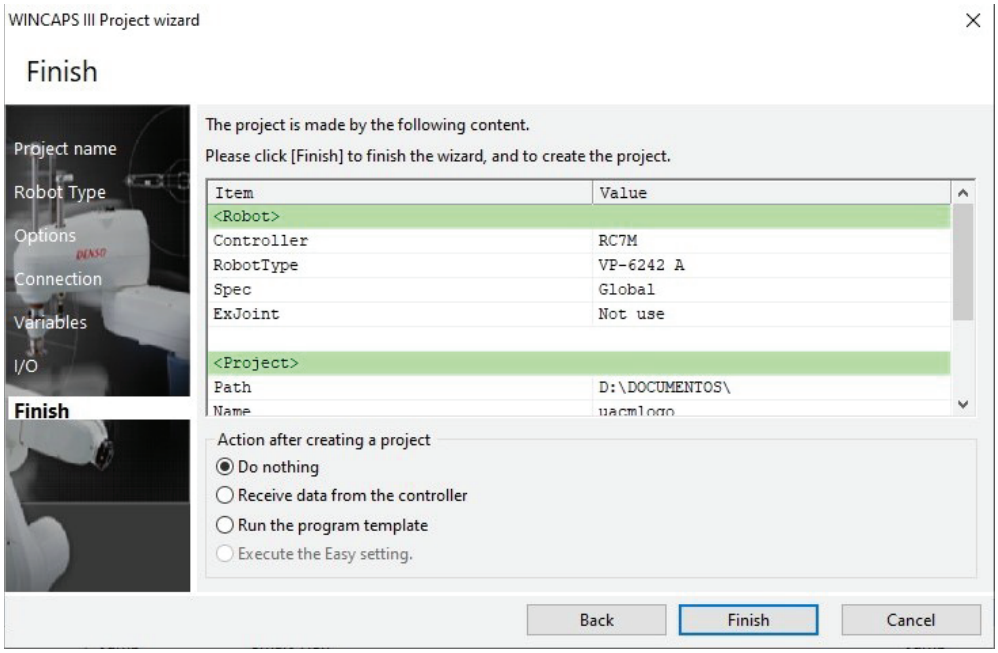


Figura D.40: Especificaciones del proyecto en Wincaps iii

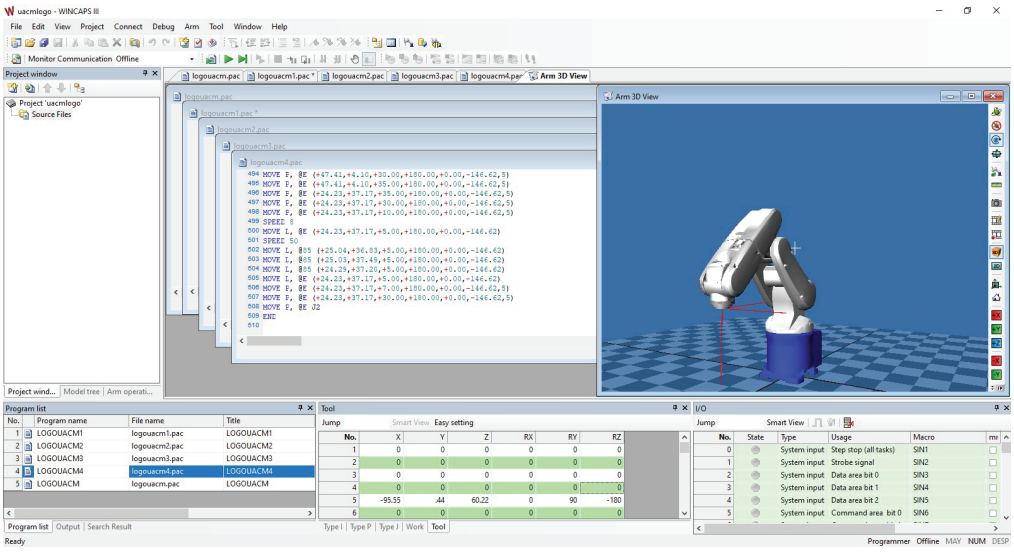
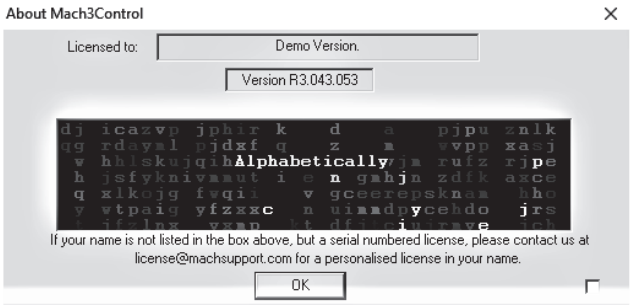


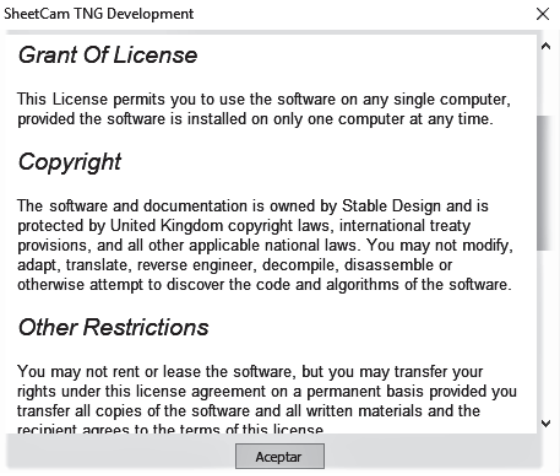
Figura D.41: Entorno de trabajo en Wincaps iii

Apéndice E
Licencias

Las licencias de softwares que se adquirieron fueron las de los softwares MasterCAM y RobotMaster. Los softwares SheetCam, Wincaps iii, Mach 3, Eagle, fueron versiones de prueba las utilizadas.



(a) ***Licencia Mach3***

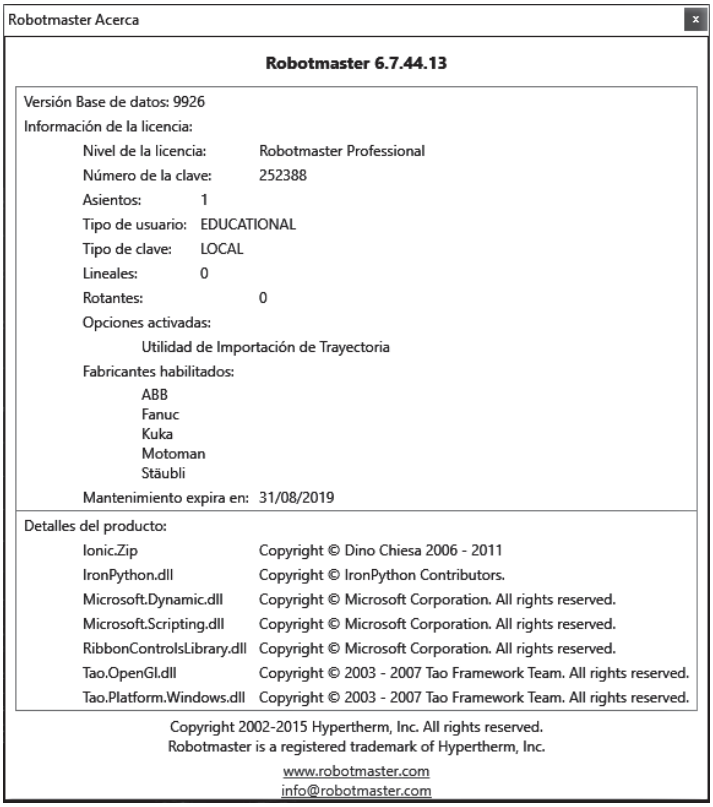


(b) ***Licencia Sheetcam***

Figura E.1: Licencias de prueba



(a) ***Licencia MasterCAM***



(b) ***Licencia RobotMaster***

Figura E.2: Licencias de prueba

Bibliografía

[1] (1995). *Scorbot-ER III, Users manual*. Eshed Robotec, 6 edition. 67

[2] (2002). *Programmers manual I, Program desing and comads*. Denso Wave Incorporated. 52

[3] (2005). *General information about robot*. Denso Wave Incorporated. 63

[4] (2008). *Wincaps III guide*. Denso Wave Incorporated. 106

[5] Bajd, T., Mihelj, M., Lenarcic, J., Stanovnik, A., and Munih, M. (2010). *Robotics*, volume 43 of *Intelligent systems, control, and automation: Science and engineering*. Springer. 10, 12, 40

[6] Barrientos, A., Peñin, L. F., Balaguer, C., and Aracil, R. (2007). *Fundamentos de robótica*. McGraw-Hill, second edition. 10, 14

[7] Baturone, A. (2001). *Robótica manipuladores y robots móviles*. Marcombo, Barcelona. 13

[8] Craig, J. J. (2005). *Introduction to robotics: Mechanics and control*. Pearson education, third edition. 21, 31

[9] Desai, J. P. (2005). D-h convection. In Kurfess, T. R., editor, *Robotics and Automation Handbook*, chapter 8. CRC Press. 21, 23

[10] Hernandez, J. C. and Vizán, A. (2015). *Sistemas de automatización y robótica*. EOI Escuela de Organizacion Industrial, Madrid. 2

[11] Kelly, R. (1994). *Control de movimiento de robots manipuladores*. Centro de instigacion cientifica y de educacion superior de ensenada, Mexico. 39, 41

[12] Khare, R., Sapra, A., Gupta, I., Kumar, V., Khare, A., Wig, T., and Chawla, M. (2007). *Robotics, Appin Knoeledge Solutions*. Infinity Science Press LLT. 15

[13] Martinez, J., Guerrero, G., and Vela, L. G. (2006). Modelado del robot scorbot er-5+. CENIDET, Congreso Nacional de Control Automatico, UNAM. 33, 38

[14] Radhakrishnan, P., Subramanyan, S., and Raju, V. (2008). *CAD / CAM / CIM*. New Age International Publisher, third edition. 19

[15] Reyes, F. (2011). *Robótica. Control de robots manipuladores*. Alfaomega, México, firts edition. 31, 35

[16] Saha, S. K. (2008). *Introduction to robotics*. McGraw-Hill, India. 28

[17] Sandin, P. E. (2003). *Robots mechanisms and mechanical devices*. McGraw-Hill. 16

[18] Siciliano, B., Sciavicco, L., Villani, L., and Oriolo, G. (2009). *Robots Modelling, planing and control*. Springer-Verlag. 21

[19] Wolf, A., Stelnmann, R., and Schunk, H. (2005). *Grippers in motion*. Springer-Verlag, Berlin. 17