



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

**PROGRAMA DE MAESTRÍA Y DOCTORADO EN
INGENIERÍA**

FACULTAD DE INGENIERÍA

**DESARROLLO DE UN SISTEMA DE
CONTROL PARA UNA MICROMÁQUINA
HERRAMIENTA BASADO EN CÓDIGO
ESTÁNDAR PARA CNC**

T E S I S

QUE PARA OBTENER EL GRADO DE:

MAESTRO EN INGENIERÍA

ELÉCTRICA - SISTEMAS ELECTRÓNICOS

P R E S E N T A:

HÉCTOR HUGO SILVA LÓPEZ, ING.

TUTORES:

DR. ALBERTO CABALLERO RUIZ.
DR. LEOPOLDO RUIZ HUERTA.



CIUDAD UNIVERSITARIA SEPTIEMBRE, 2008.

JURADO ASIGNADO:

Presidente : M. I. Escobar Salguero Larry

Secretario : Dr. Ruiz Huerta Leopoldo

Vocal : Dr. Caballero Ruiz Alberto

1^{er}. Suplente: M. I. Quintana Thierry Sergio

2^{do}. Suplente: M. I. Damián Zamacona Ricardo

Lugar donde se realizó la tesis:

Centro de Ciencias Aplicadas y Desarrollo Tecnológico, CCADET.
Grupo de Micromecánica y Mecatrónica, GMM.

TUTOR DE TESIS:

Dr. Caballero Ruiz Alberto.

FIRMA

AGRADECIMIENTOS

Este trabajo se desarrolló en el Centro de Ciencias Aplicadas y Desarrollo Tecnológico, UNAM bajo la tutela del Dr. Alberto Caballero Ruiz y el Dr. Leopoldo Ruiz Huerta (Grupo de Micromecánica y Mecatrónica). El trabajo fue apoyado por los proyectos CIAM-CONACyT2005-C02-51843/A-1 CONACyT 60895, CONACyT 52416, CONACyT 52250 y CONACyT 50231.

AGRADECIMIENTOS

A mi padre Antonio y a mi tío Gustavo, quienes me brindaron su apoyo y recursos de manera incondicional durante toda mi formación académica y social.

A mi abuela Irene y a mi madre María Elena, quienes con sus palabras y acciones me demuestran cada día su cariño y apoyo en cualquier momento de mi vida.

A mis hermanos Juan, Verónica y Yadira, quienes han motivado para bien, cada una de mis acciones no solo en este proyecto sino también en lo que llevo de vida. Los amo hermanos.

A Angélica, quien con su amor, amistad, consejos y paciencia me ha demostrado que no hay mejor camino para conseguir lo que uno quiere que preparándose día con día. Gracias Amor por formar parte de mi familia.

A mis tutores y amigos, Alberto y Leopoldo, quienes con su apoyo, consejos y enseñanzas han fomentado mi superación profesional, además de significar un portal para nuevas experiencias a nivel personal.

A Germán, por tu entrañable hermandad, conocimientos, trabajo, consejos y buenos momentos durante toda mi estancia en el grupo. Gracias hermano.

A mis amigos Mauricio, Urso, Jesús, Mary, Amparo, Ares y Ricardo, quienes con su presencia y consejos me han alentado a seguir perseverando en mi desarrollo profesional.

A mis sinodales Sergio, Ricardo y Larry, quienes con sus enseñanzas y desempeño han significado un excelente ejemplo en lo personal y profesional.

A la Universidad, por confiar en mí al darme los recursos y la oportunidad de formar parte de un mundo lleno de conocimientos e inolvidables experiencias.

CONTENIDO.

INTRODUCCIÓN	1
1. ANTECEDENTES	3
1.1 Micromecánica en el mundo.....	5
1.2 Micromecánica en México.....	7
1.3 Sistemas de control implementados en el GMM..... anteriores a este trabajo.	10
1.4 Sistemas CNC.....	13
1.4.1 Definición de CNC.....	13
1.4.2 Clasificación del CNC.....	13
1.4.3 Ventajas y Desventajas del CNC.....	14
1.4.4 Programación y códigos.....	15
2. OBJETVO Y ALCANCES	17
2.1 Definición del problema.....	17
2.2 Objetivo.....	19
2.3 Alcances.....	19
3. MARCO TEÓRICO	20
3.1 Introducción a la programación en sistemas CNC.....	21
3.2 Programación en sistemas CNC.....	21
3.2.1 Estructura de un programa para CNC.....	21
3.2.2 Principales códigos de programación CNC.....	22
3.3 Técnicas de control en Sistemas CNC.....	23
3.3.1 Técnica por referencia de pulsos.....	23
3.3.2 Técnica por muestreo de datos.....	23

3.4 Interpolación por DDA.....	24
3.4.1 Interpolación Lineal.....	24
3.4.2 Interpolación Circular.....	26
3.5 Interpolación Incremental (ó por escalones).....	27
3.6 Interpolación por Búsqueda Directa.....	29
4. DESARROLLO DEL SISTEMA DE CONTROL.....	31
4.1 Caso de estudio.....	31
4.2 Características esperadas.....	33
4.3 Propuesta de control.....	33
4.3.1 Características del sistema de control.....	35
4.3.1a Modos de operación.....	36
4.3.1b Parámetros monitoreados.....	37
4.3.2 Códigos G y M soportados.....	37
4.3.2a Descripción y sintaxis de los códigos.....	38
4.4 Simulación de los métodos de Interpolación.....	38
4.5 Evaluación de los métodos de Interpolación.....	42
4.6 Selección del método de Interpolación.....	43
4.7 Programa de control.....	44
4.8 Interfaz computacional.....	46
4.9 Ejemplos del proceso de control.....	48
4.9.1 Ejemplo (a).....	49
4.9.2 Ejemplo (b).....	51
4.10 <i>Hardware</i> electrónico.....	52
4.10.1 Circuito PBL3717A.....	52
4.10.2 Circuito A3988.....	55

ÍNDICE DE FIGURAS.

Figura 1. Desarrollo de la Micromecánica en el GMM [8].....	4
Figura 2. Microrobots como ayudantes en intervenciones..... quirúrgicas.	5
Figura 3a. Microrobot para tareas de inspección [32].....	6
Figura 3b. Microrobot volador [33].....	6
Figura 4a. Microtorno japonés.....	6
Figura 4b. Microcentro de maquinado ucraniano.....	6
Figura 4c. Microfábrica japonesa automatizada.....	6
Figura 5. Primer prototipo de MMH fabricado en México.....	7
Figura 6. Segundo prototipo de MMH fabricado en México.....	8
Figura 7. Motores a pasos desarrollados en el GMM.....	9
Figura 8. Motor a pasos de alta potencia (de doble estator) [10].....	9
Figura 9. Software de control empleado en el primer prototipo.....	10
Figura 10. Micropiezas fabricadas con el primer prototipo [29].....	11
Figura 11. Recursos empleados por la aplicación computacional.....	11
Figura 12. Pantalla principal de la interfaz computacional para..... el control de MMH [12].	12
Figura 13. Resultados obtenidos por la aplicación computacional..... programada en <i>Visual Basic</i> [12].	12
Figura 14. Componentes básicos de un CNC.....	13
Figura 15. Proceso de manufactura en un CNC [37].....	15
Figura 16. Diagrama general de un sistema CNC.....	18
Figura 17. Bloques fundamentales de un controlador CNC.....	20
Figura 18. Sistema por referencia de pulsos.....	23
Figura 19. Sistema por referencia de palabra.....	24

Figura 20. Algoritmo DDA para segmentos rectilíneos.....	25
Figura 21. Algoritmo DDA para segmentos circulares (1er cuadrante).....	26
Figura 22. Algoritmo por interpolación incremental.....	28
Figura 23. Algoritmo por el método de Búsqueda Directa.....	30
Figura 24. Nueva versión del primer prototipo de MMH..... desarrollado en México.	32
Figura 25. Motor a pasos bipolar de magneto permanente.....	32
Figura 26. Fresadora CNC del GMM [6].....	34
Figura 27. Sistema de control propuesto.....	34
Figura 28. Pantalla del programa que simula los métodos..... de interpolación.	39
Figura 29. Simulación del método por escalones.....	39
Figura 30. Simulación del método por DDA.....	40
Figura 31. Simulación por el método de Búsqueda Directa.....	41
Figura 32. Simulación de los tres métodos de interpolación.....	42
Figura 33. Acercamientos a las zonas 1, 5 y 2 de la figura 32.....	42
Figura 34. Rutinas y sub-rutinas principales del programa..... de control.	45
Figura 35. Pantalla de inicio de la interfaz computacional.....	46
Figura 36. Secciones de la interfaz computacional.....	47
Figura 37. Proceso de control para la cadena "JX+;".....	49
Figura 38. Proceso de control para un código G02.....	51
Figura 39. Circuito interno del PBL3717A.....	53
Figura 40. Circuito para controlar un motor a pasos bipolar..... de manera bidireccional.	54
Figura 41. Circuito interno del A3988 para controlar dos..... motores a pasos bipolares.	56

Figura 42. Sistema de potencia para dos actuadores de la MMH.....	58
Figura 43. Sistema propuesto para las pruebas al sistema..... de control.	61
Figura 44. Comparador óptico del Laboratorio de Metrología.....	61
Figura 45. Señales generadas por la tarjeta de control para..... los tres ejes de la MMH.	62
Figura 46. Señales generadas por la etapa de potencia (sin carga).....	62
Figura 47. Señales generadas por la etapa de potencia..... con carga inductiva.	63
Figura 48. Circuito LC para reducir efectos inductivos en..... las fases del motor.	64
Figura 49. Señales generadas por el sistema de potencia con..... circuito LC.	64
Figura 50. Señales generadas con circuito LC de segundo..... orden en fase 1.	65
Figura 51. Señales generadas por un código G02.....	65
Figura 52. Maquinados para determinar el juego mecánico en..... toda el área de trabajo.	66
Figura 53. Maquinados para corroborar la compensación del..... juego mecánico.	67
Figura 54. Maquinados con interpolación circular en..... ejes X y Y.	69
Figura 55. Errores en la compensación del juego mecánico..... (Extremos izquierdo y derecho).	70
Figura 56. Errores en la compensación del juego mecánico..... (Polos superior e inferior).	71
Figura 57. Escudo Puma hecho en <i>Solid Works</i>	73
Figura 58. Proceso de manufactura del Puma en <i>VisiCAM</i>	73
Figura 59. Código G y M generado en <i>VisiCAM</i>	74
Figura 60. Pantalla del proceso de maquinado de..... un logo de los Pumas.	75

Figura 61. Puma maquinado con interfaz computacional [6].....	75
(izq.), y Puma maquinado con sistema de control mediante Microcontrolador PIC (der.).	
Figura 62. Proceso de maquinado del logo de los Pumas de 5mm.....	76
Figura 63. Maquinado continuo de un Puma de 1.2mm (izq.),.....	77
y Puma maquinado en espejo de 1.2mm(der.).	
Figura 64. Proceso de maquinado del Puma de 1.2mm.....	78
Figura 65. Dos maquinados de un Puma de 1mm.....	78
Figura 66. Puma circunscrito en circunferencia de 1mm.....	78
de diámetro.	
Figura 67. Logotipo del GMM hecho en <i>Solid Works</i>	79
Figura 68. Proceso de manufactura del logotipo del GMM en.....	80
<i>VisiCAM</i> .	
Figura 69. Pantalla del proceso de maquinado del logotipo.....	80
del GMM de 5mm.	
Figura 70. Dos maquinados del logotipo del GMM de 12mm.....	81
(izq.) y 5mm (der.).	
Figura 71. Acercamientos del maquinado de la figura 70 (der.).....	81
Figura 72. Maquinado del logotipo del GMM de 1mm.....	83
Figura 73. Esquemático del sistema de potencia para.....	102
dos motores mostrado en la figura 42.	
Figura 74. Cara inferior del circuito impreso para el.....	103
sistema de potencia mostrado en la figura 42.	
Figura 75. Cara posterior del circuito impreso para el.....	104
sistema de potencia mostrado en la figura 42.	

ÍNDICE DE TABLAS.

Tabla 1. Clasificación de un CNC [36].....	14
Tabla 2. Ventajas y desventajas de un CNC [36].....	14
Tabla 3. Variables de la programación CNC [37].....	22
Tabla 4. Características de los Microcontroladores.....	34
Tabla 5. Características de la tarjeta de control.....	35
Tabla 6. Recursos del Microcontrolador PIC18F8720.....	36
Tabla 7. Resultados obtenidos de las simulaciones a los algoritmos propuestos.	41
Tabla 8. Parámetros de importancia y valores relativos asignados.	43
Tabla 9. Evaluación de los algoritmos de interpolación.....	44
Tabla 10. Secuencias para los diferentes modos de operación del PBL3717A.	55
Tabla 11. Parámetros de importancia en la comparación de circuitos de potencia.	57
Tabla 12. Evaluación de los circuitos de potencia.....	57
Tabla 13. Magnitudes del juego mecánico para los ejes X y Y.....	67
Tabla 14. Resultados de los maquinados mostrados en la figura 53.	68
Tabla 15. Resultados de los maquinados mostrados en la figura 54.	70
Tabla 16. Dimensiones, juego mecánico y error relativo del Puma de 5mm.	76
Tabla 17. Dimensiones y juego mecánico del Puma de 1.2mm	77
Tabla 18. Dimensiones y juego mecánico del Puma de 1mm.....	79
Tabla 19. Mediciones realizadas al logotipo del GMM de 5mm.....	82
Tabla 20. Mediciones realizadas al logotipo del GMM de 1mm.....	83

Tabla 21. Conjunto de códigos “G” a interpretar.....	88-91
Tabla 22. Conjunto de códigos “M” a interpretar	91-93
Tabla 23. Modos de operación de la interfaz computacional.....	93-94

Introducción

Actualmente, el hombre cuenta con una gran diversidad de sistemas mecánicos automatizados capaces de resolver un alto porcentaje de todas sus necesidades. El nivel de automatización de estos sistemas es cada día más complejo, logrando desarrollar tareas cada vez más complicadas, sin la intervención de la mano del hombre, representando éstos una tremenda inversión a las industrias y otros campos, en cuanto a costo, espacio y consumo energético.

Un campo de estudio, la Micromecánica, propone desarrollar sistemas mecánicos de pequeñas dimensiones automatizados. El desarrollo de este campo de estudio ha llevado a la creación de sistemas cada vez más compactos de alta eficiencia, bajo costo, y de menor consumo energético, ocupando espacios cada vez más reducidos [1].

Así pues, la Micromecánica propone que mediante el empleo de las técnicas de la mecánica convencional, se lleve a cabo la construcción de dispositivos de dimensiones inferiores al milímetro que tengan aplicación en diversos campos de la industria, medicina, biotecnología, industria química, aeroespacial, etc. [2].

La propuesta de generar microdispositivos mecánicos mediante el uso de técnicas de la mecánica convencional surge primeramente en Japón y Ucrania a mediados de los 90's. En los inicios de estas propuestas, los investigadores se dieron a la tarea de desarrollar microequipo que permitiera comprobar las hipótesis generadas en este campo de estudio. Fue a partir de este momento, cuando surge un interés a nivel mundial, y países como Alemania, Corea, Noruega, Suiza, EUA y México muestran interés por este campo [3].

En México, el Grupo de Micromecánica y Mecatrónica (GMM) a través del Centro de Ciencias Aplicadas y Desarrollo Tecnológico (CCADET), en la Universidad Nacional Autónoma de México (UNAM), propone como principal línea de investigación *Generar tecnología de punta para la fabricación de piezas con dimensiones inferiores al milímetro* con el objetivo de desarrollar tecnologías de producción totalmente automatizadas, enfocadas a la creación de equipo e instrumentación de bajo costo y alta eficiencia [4].

El principio básico propuesto por el GMM, consiste en el desarrollo de generaciones de microequipo, enfocado a la manufactura de microdispositivos, en donde cada generación tiene dimensiones menores a la anterior. La primera generación de microequipo es desarrollada con Máquinas Herramienta (MH's) de tamaño convencional, mientras que el microequipo de la siguiente generación es desarrollado con ayuda del microequipo de la generación que lo precede [4].

Dentro de la principal línea de investigación del GMM, se tienen diversas líneas de trabajo, una de ellas es el desarrollo de sistemas de control que permitan automatizar los procesos de manufactura, con el fin de generar el microequipo de las siguientes generaciones. Por tal razón, es necesario realizar un estudio enfocado a las técnicas de control empleadas en los sistemas de control de equipos convencionales.

A continuación, se presenta un trabajo en donde se analizan y evalúan tres algoritmos de control para el desarrollo de un sistema de control, para una micromáquina herramienta (MMH) de primera generación, con la capacidad de interpretar código estándar empleado en centros de maquinado convencionales dotados de control numérico asistido por computadora (CNC).

El contenido de este trabajo se desglosa en cinco capítulos. En el *capítulo 1*, se tratan los antecedentes generales del estado actual de la Micromecánica, su desarrollo en México, así como las diferentes propuestas de control que se han implementado, en el GMM, hasta antes de este trabajo.

Dentro del *capítulo 2*, se define el problema a resolver así como el objetivo y los alcances de este trabajo. Mientras que en el *capítulo 3*, se da un marco teórico acerca de los sistemas CNC, conceptos, clasificación, ventajas, desventajas, programación, algoritmos de control utilizados en la automatización de MH's, así como las características generales del prototipo a controlar.

Por otra parte, en el *capítulo 4* se describe el desarrollo de este trabajo, estableciendo métodos de selección para el algoritmo de control y los componentes electrónicos a emplear. Terminando con una descripción de la interfaz computacional desarrollada para el monitoreo del sistema.

Por último, en el *capítulo 5* se describen las pruebas realizadas al *hardware* electrónico y sistema micromecánico en cuestión. Además, se tiene una sección de conclusiones y trabajo a futuro en la cual se pretende analizar los logros alcanzados con este trabajo, y por supuesto, proponer mejoras al sistema desarrollado con base en la experiencia obtenida.



Antecedentes

Actualmente a nivel mundial, se aprecia una tendencia en el desarrollo de nuevas tecnologías enfocadas a la miniaturización de los sistemas, ejemplos de estas tecnologías son los *MicroElectroMechanical Systems* (MEMS), desarrollados principalmente en los Estados Unidos de América, *MicroSystems Technology* (MST), cuyo mayor auge está en los países europeos; y por último los *MicroMachine Technology* (MMT) desarrollados en Japón [5].

El principal interés en el desarrollo de nuevas tecnologías para la fabricación de elementos micromecánicos, obedece a las restricciones causadas por los materiales y geometrías de las piezas [1]. Como ya se ha mencionado anteriormente, el motivo de hacer uso de las técnicas de la mecánica convencional para la obtención de micromecanismos tiene como fin adoptar procedimientos conocidos que nos permitan obtener las tolerancias requeridas.

En el diagrama de la figura 1, se muestra (en naranja) el camino que el GMM ha seguido desde el año 1999 para el desarrollo de microequipo. Por un lado tenemos la tecnología de los MEMS, la cual puede ser utilizada para crear dispositivos de bajas dimensiones con componentes mecánicos simples generados a partir de técnicas de fabricación de circuitos integrados, sin embargo estos dispositivos poseen prácticamente dos dimensiones debido a su escaso espesor [7].

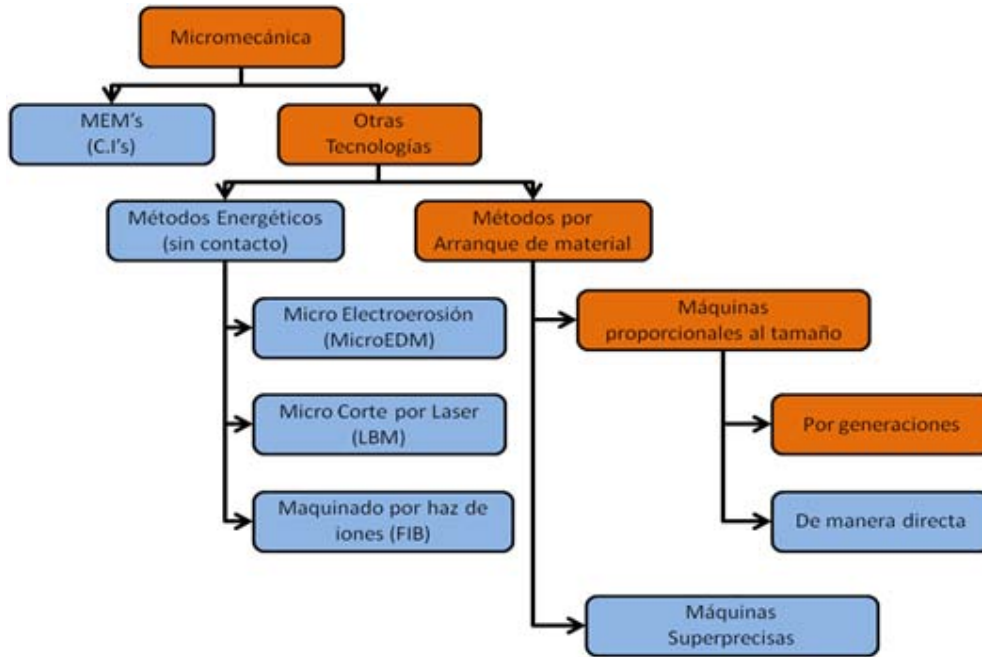


Figura 1. Desarrollo de la Micromecánica en el GMM [8].

Debido a que para el desarrollo de sistemas micromecánicos (basados en mecánica convencional), como: MMH, micromanipuladores y microrobots, se requieren de estructuras mecánicas con las siguientes características [5]:

- Tres dimensiones reales.
- Formas complejas.
- Posibilidades de ensamble y movimiento.
- Uso de diferentes materiales.

Esto nos deja ver que es necesario buscar otro camino para el desarrollo de microdispositivos complejos. Así pues, otras tecnologías existentes difieren en el aspecto de desbaste, dividiéndose básicamente en métodos energéticos (sin contacto) y por arranque de material. Actualmente, los prototipos de MMH manufacturados en el GMM, adoptan la metodología por arranque de material, sin implicar esto que no se esté interactuando con métodos energéticos.

Por lo tanto, para generar microequipo se tienen dos alternativas a seguir: la primera de ellas consiste en el empleo de máquinas superprecisas, mientras que la segunda se basa en la disminución de tamaño de MH y manipuladores [6].

El empleo de maquinaria superprecisa no es del todo recomendable para una aplicación comercial, ya que requiere de una gran inversión debido a que su costo es elevado. Mientras que, la disminución en tamaño de MH y manipuladores, resulta en un decremento en los costos de producción de dispositivos comerciales [7].

Como se mencionó en la introducción de este trabajo, el principio básico propuesto por el GMM, para generar microequipo a bajo costo, es desarrollar generaciones de microequipo. En consecuencia, el GMM desarrolló máquinas generacionales, con dimensiones proporcionales a la pieza de trabajo.

Esta última decisión se basa en el método conocido como Tecnología de Microequipo (MET), el cual consiste en desarrollar microdispositivos generacionales; es decir, se desarrollan máquinas que posteriormente se utilizarán para el desarrollo de generaciones subsecuentes de menor tamaño [5].

1.1 Micromecánica en el mundo.

El auge por el desarrollo de microdispositivos se inició a mediados de los años 80's; por ejemplo, en el campo de la medicina la demanda de microequipo aumenta con aplicaciones en microscopía, diagnóstico, intervenciones quirúrgicas poco invasivas, tratamientos, etc. La reducción de dimensiones en el instrumental médico permite que la inspección en el cuerpo humano sea menos riesgosa, así mismo, eleva la precisión y eficiencia en los diagnósticos [7].



Figura 2. Microrobots como ayudantes en intervenciones quirúrgicas [31].

Como ejemplos de las diversas aplicaciones de los sistemas micromecánicos, están dos pequeños robots, representados en la figura 2, destinados a la inspección en ojos (izquierda) y espalda (derecha) [31]. Además, de tener un uso potencial en labores donde un error del hombre sería crítico. En la figura 3a, se muestran un microrobot para tareas de inspección desarrollado por laboratorios *Sandia* de 40x19x17mm con menos de 30gr de peso [32]. Por otro lado, en la figura 3b, se muestra el microrobot volador desarrollado por *Seiko-Epson Co.* de 85mm de alto por 136mm de diámetro con un peso de 13gr y autonomía de vuelo por 3 minutos [33].

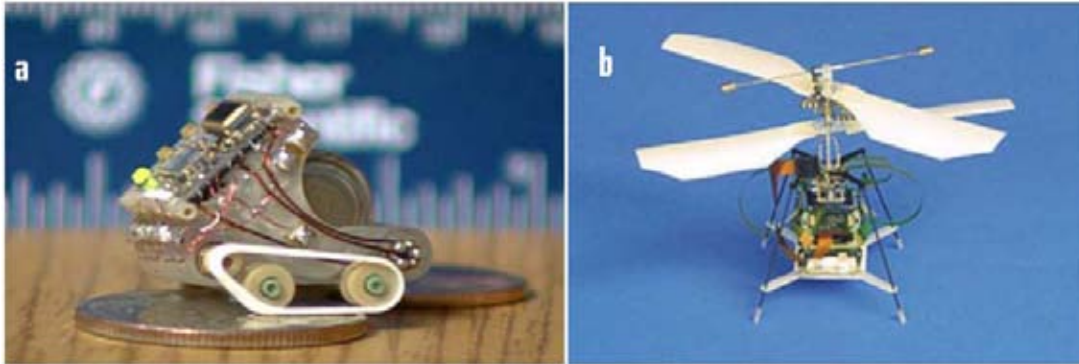


Figura 3. (a) Microrobot para tareas de inspección [32] (b) Microrobot volador [33].

En lo que respecta al desarrollo de microequipo como: MMH, definidas éstas en el GMM como sistemas extremadamente pequeños que deben su nombre no solo a su tamaño, sino también al tamaño de sus componentes, y ampliando la definición, al tamaño de las piezas fabricadas con éstas [4], resultan de gran importancia para el campo de la manufactura de pequeños productos industriales.

Como ejemplo de MMH desarrolladas a nivel mundial, en la figura 4 se muestran tres desarrollos importantes. En la figura 4a, se muestra un microtorno desarrollado en el *Mechanical Engineering Laboratory* del *National Institute of Advanced Industrial Science and Technology*, de dimensiones 30x30x30mm, desarrollado en Japón en el año de 1996 [9]. En la figura 4b, se muestra un desarrollo (un año más tarde) del *International Research and Training Center of UNESCO/IIP of Information Technologies and Systems, National Academy of Science of Ukraine* de un prototipo de microcentro de maquinado [34]. Mientras que en la figura 4c, se muestra una microfábrica automatizada desarrollada en el año 2000 por investigadores del *Mechanical Engineering Laboratory* del *National Institute of Advanced Industrial Science and Technology*, en Japón [35].

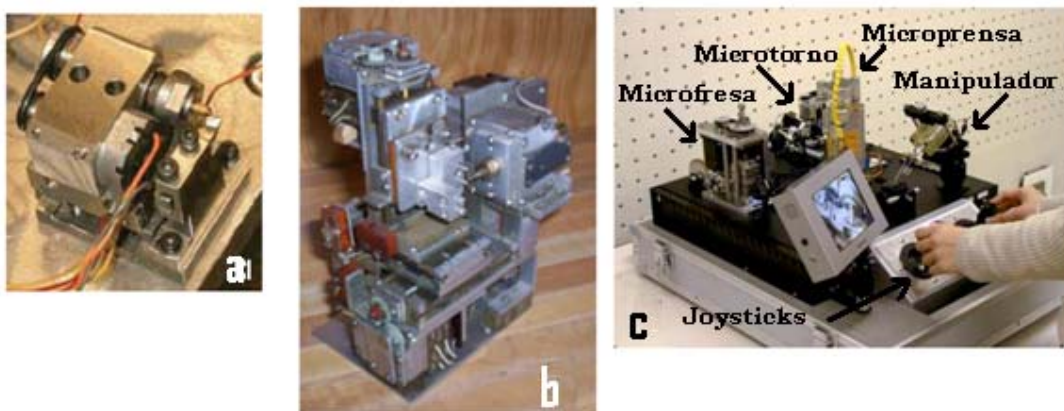


Figura 4. (a) Microtorno japonés, (b) Microcentro de maquinado ucraniano, (c) Microfábrica japonesa automatizada.

Por otra parte, a nivel nacional se han generado avances importantes en el campo de la Micromecánica, y los cuales se describirán en el siguiente apartado.

1.2 Micromecánica en México.

En México, el GMM en su línea de investigación referente al desarrollo de tecnología de punta para la fabricación de micropiezas, da inicio a la creación de microequipo generacional presentando en el año 2000 (dentro del *First International Conference on Mechatronics and Robotics* en San Petersburgo, Rusia) el primer prototipo de MMH, mostrado en la figura 5, con capacidad de realizar procesos de torneado, fresado y taladrado. En consecuencia, se da inicio a la primera generación de equipo micromecánico automatizado de bajo costo desarrollado totalmente en México [1].

Las principales características de esta MMH son: dimensiones de 130x160x85mm, tres grados de libertad que permiten desplazamientos lineales y uno rotacional, como actuadores se utilizan motores a pasos desarrollados en la misma entidad, resolución de 1.87 μ m lineales por paso del motor, y capacidad para producir piezas en un intervalo de los 100 μ m a los 5mm.

Como resultado de la experiencia que se obtuvo durante el diseño y construcción del prototipo de la figura 5, el GMM, en el año 2005 presenta su segundo prototipo de MMH de primera generación, basado en mecanismos de palancas y paralelogramos, lo cual permite que los movimientos desarrollados tengan una mejor resolución pero generando como movimiento secciones circulares.

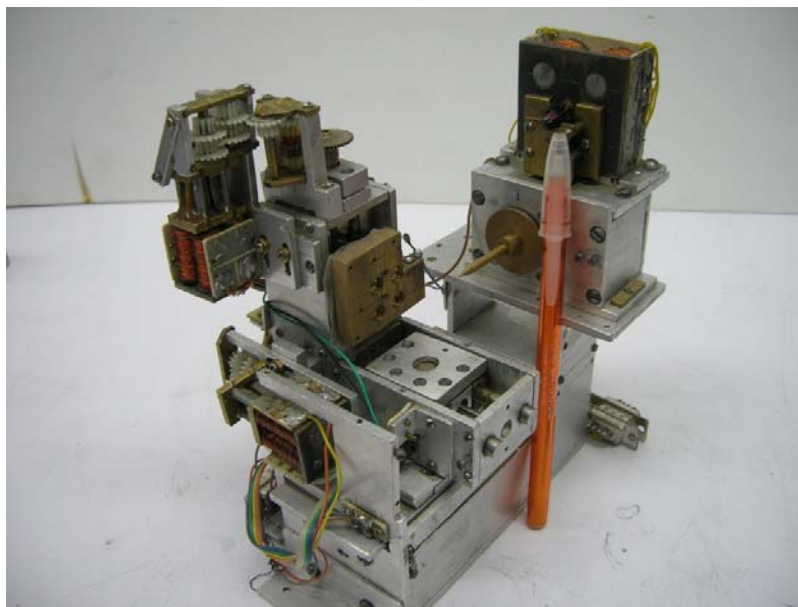


Figura 5. Primer prototipo de MMH fabricado en México.

Este segundo prototipo mostrado en la figura 6, tiene dimensiones de 180x210x135mm, cuatro grados de libertad, los actuadores empleados son motores a pasos, la resolución lineal es de $0.596\mu\text{m}$ y tiene la capacidad de producir piezas en un intervalo de $50\mu\text{m}$ hasta 5mm. Este prototipo está diseñado para operar mediante el control de un sistema de cómputo convencional.



Figura 6. Segundo prototipo de MMH fabricado en México.

Dentro de la principal línea de investigación del GMM, se encuentran en desarrollo varias actividades, dos de las más importantes son: el diseño y manufactura de micromotores a pasos, y el desarrollo de sistemas de control para la automatización de procesos de manufactura.

Con la construcción de sus propios micromotores a pasos el GMM busca la generación de conocimiento al respecto, y la reducción de costos en el desarrollo de sus prototipos micromecánicos. Si bien es cierto que el costo inicial de desarrollo puede ser alto debido a la curva de aprendizaje, también es cierto que el uso de una metodología puede mejorar el proceso de desarrollo de los micromotores a pasos, y con ello disminuir el costo de fabricación de los mismos [10]. En las figuras 7 y 8, se pueden apreciar los diseños más representativos de motores a pasos que se han desarrollado en el GMM, junto con objetos (monedas, lapiceros y plumas) que permiten dar una idea de las dimensiones de los mismos.

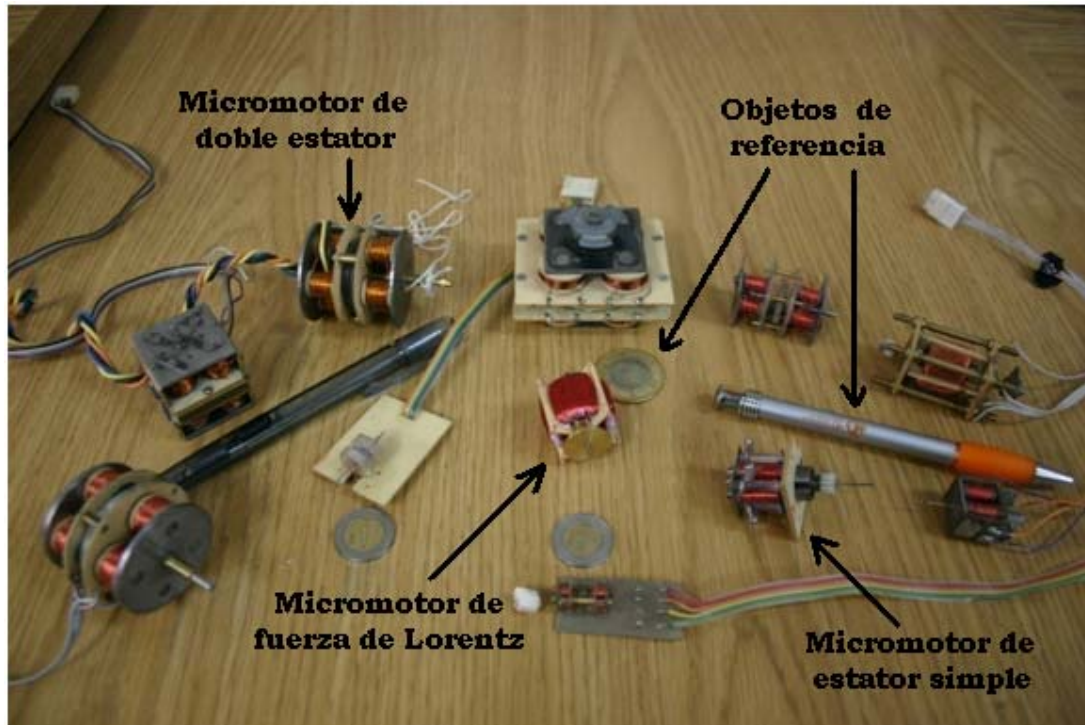


Figura 7. Motores a pasos desarrollados en el GMM.

El motor a pasos más grande desarrollado en el GMM, se aplicó en el campo de las ciencias médicas, al desarrollar un sistema duplicador de pulso cardíaco para la evaluación de bioprótesis cardíacas [11], para el que se fabricó un motor de alta potencia, cuyas dimensiones son significativamente mayores (del orden de 15 ó 20 veces mayor en tamaño) a los empleados en el prototipo de MMH de la figura 5 [10].

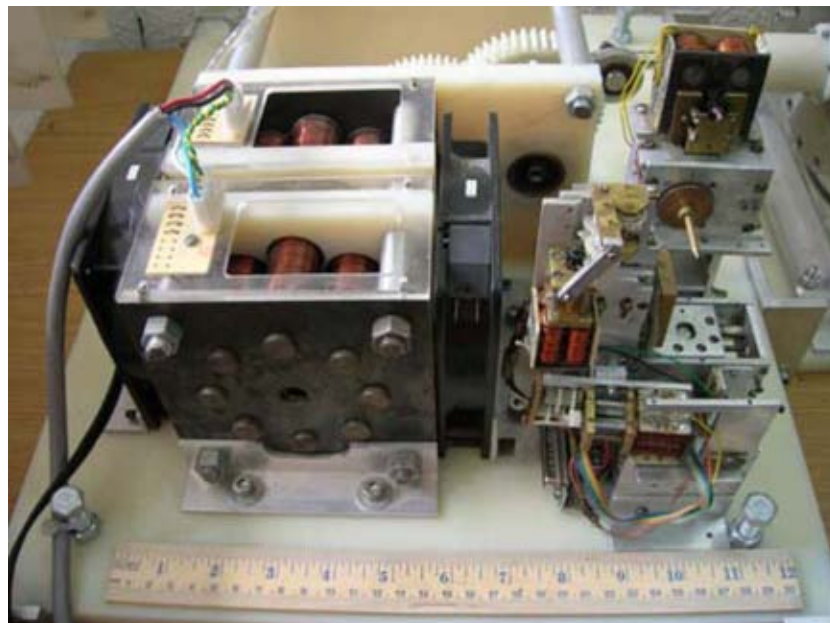


Figura 8. Motor a pasos de alta potencia (de doble estator) [10].

1.3 Sistemas de control implementados en el GMM anteriores a este trabajo.

Actualmente, el GMM cuenta con dos sistemas de control que han arrojado resultados valiosos para la automatización de equipo micromecánico [4], [12]. El primer sistema de control implementado para el prototipo de MMH de la figura 5, fue diseñado con el objetivo de controlarlo mediante una computadora personal (PC), eliminando en medida de lo posible el hardware de control externo. El sistema está compuesto de un amplificador de DC de 16 canales, y cuatro acondicionadores de señal para los sensores de contacto colocados al principio de carrera de cada eje [4].

La MMH cuenta con un conector DB25, el cual contiene las conexiones a los motores y sensores de posición de origen mediante contacto físico, y posición de materia prima por contacto eléctrico. Además, el *software* de control empleado, mostrado en la figura 9, fue desarrollado en Borland C++, y es mediante esta herramienta que se monitorea el estado del sistema micromecánico a través de dos puertos paralelos que se encuentran divididos en tres secciones, dos de ellas dedicadas al movimiento de los actuadores y la última a las señales de retroalimentación provenientes de los sensores de posición por contacto físico y por contacto eléctrico [4].

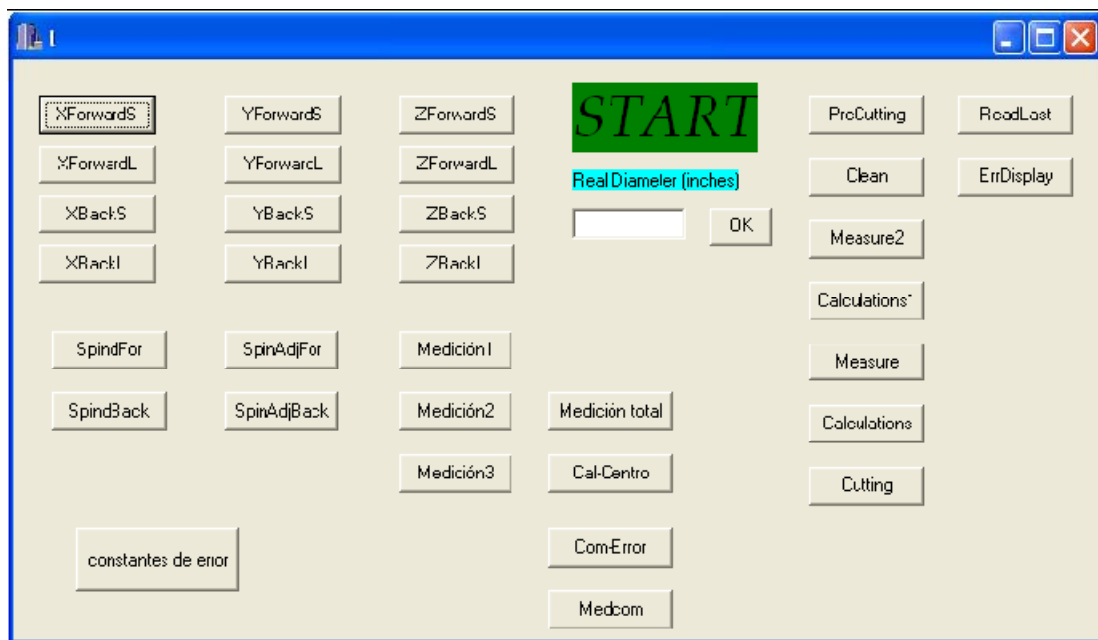


Figura 9. Software de control empleado en el primer prototipo.

Los resultados obtenidos con este sistema de control, son los mostrados en la figura 10. En donde podemos apreciar las piezas fabricadas de forma automática con este sistema de control.

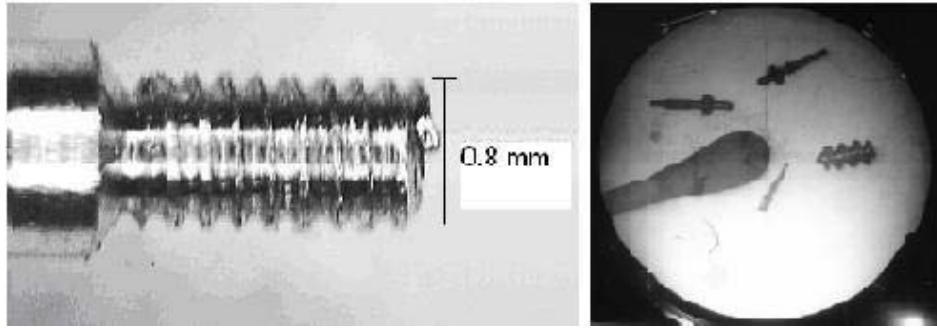


Figura 10. Micropiezas fabricadas con el primer prototipo [29].

El segundo sistema de control, y más reciente, empleado en el prototipo de MMH de la figura 5, lo conforma una aplicación computacional desarrollada en Visual Basic [12], la cual es capaz de interpretar código estándar para CNC. Este sistema de control hace uso de los mismos recursos (en cuanto a *hardware*) que el primer sistema de control, mostrados en la figura 11.

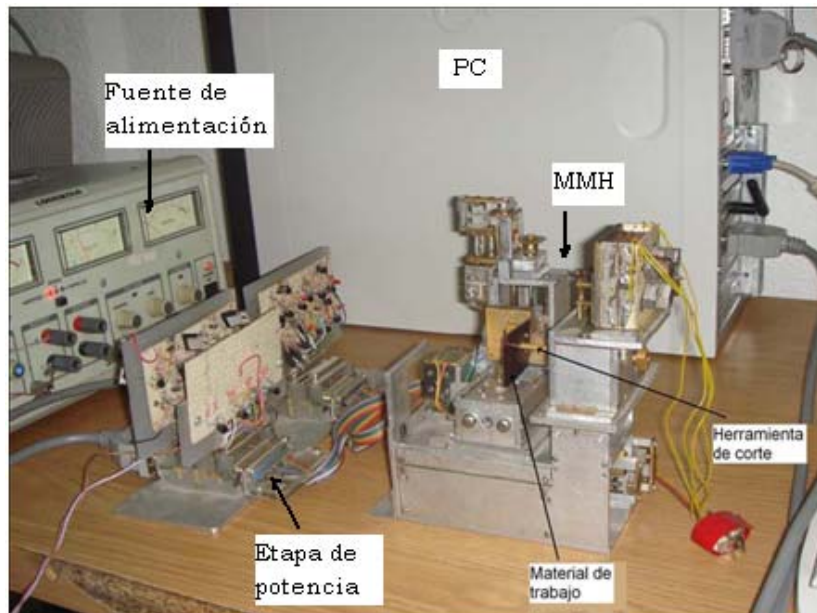


Figura 11. Recursos empleados por la aplicación computacional.

El procedimiento para manufacturar una pieza en este sistema de control, comienza después de haber generado el dibujo en un software tipo CAD, y por medio de un software CAM generar el código correspondiente para la manufactura, dicho código se importa por medio de la aplicación que se muestra en la figura 12.

Los resultados de esta aplicación computacional se muestran en la figura 13, los cuales consistieron básicamente en maquinados de figuras simples, figuras en donde se empleaban interpolaciones en dos ejes, y la ejecución de dos programas para CNC, que representaban el escudo de los pumas con dimensiones resultantes de 4mm y el logo del GMM dimensionado en 12 mm.

El desarrollado detallado de la aplicación mostrada en la figura 12, es descrito en la tesis *Desarrollo de una interfaz computacional para el control de Micromáquinas-Herramienta* [12].

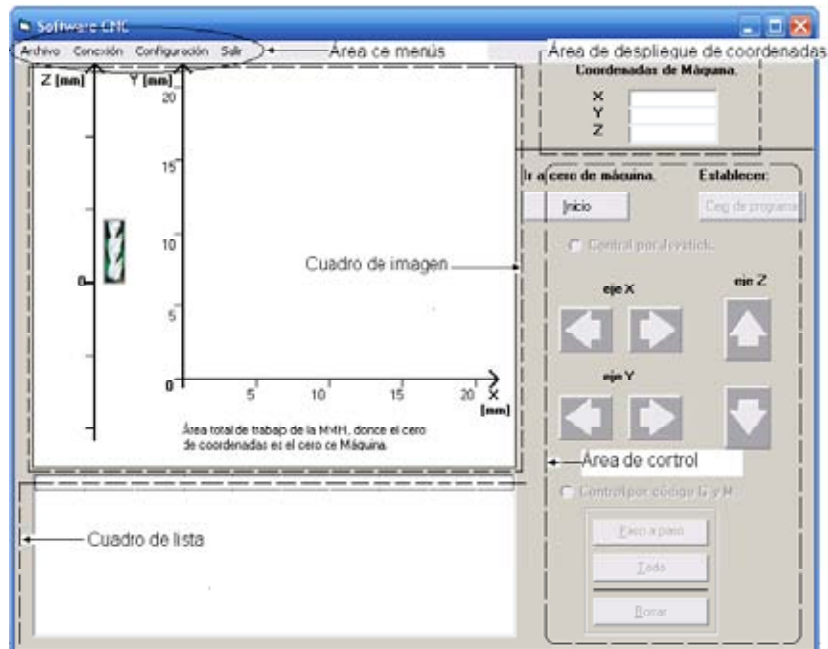


Figura 12. Pantalla principal de la interfaz computacional para el control de MMH [12].

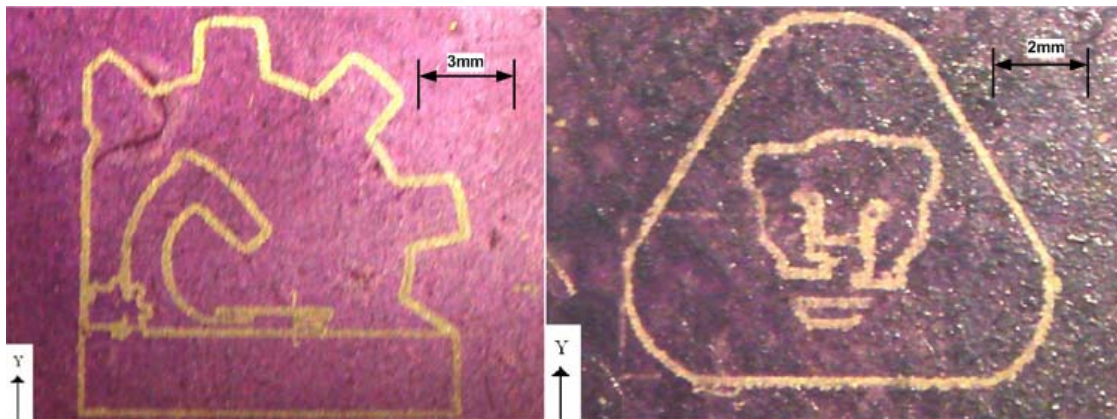


Figura 13. Resultados obtenidos por la aplicación computacional programada en *Visual Basic* [12].

En lo que respecta al segundo prototipo de MMH mostrado en la figura 6, y con el fin de tener una estimación del posible comportamiento de éste, se tiene desarrollado un programa (en plataforma *Borland C++ Builder*), con soluciones por métodos numéricos, de las ecuaciones relacionadas con las trayectorias de los casquetes semicirculares, generados por el desplazamiento en cada uno de los ejes. El análisis detallado a la solución de dichas ecuaciones se encuentra en el trabajo doctoral *Desarrollo de microequipo para células de manufactura* [1]. Actualmente, se está trabajando en una interfaz que permita controlar este segundo prototipo.

Así pues, durante el desarrollo de estos sistemas de control se han obtenido conclusiones importantes, por ejemplo, la interfaz computacional de la figura 12 presenta una ventaja fundamental en el desarrollo de micropartes respecto al primer sistema de control mostrado en la figura 9. Mientras que en el primer sistema de control el proceso de manufactura se programa en lenguaje C++, en el segundo sistema de control se reconoce código estándar para CNC. El interpretar código estándar, reduce y simplifica considerablemente el tiempo de diseño y manufactura de micropiezas.

1.4 Sistemas CNC.

Hoy en día, las máquinas de control numérico (CN) son empleadas con el fin de llevar a cabo la transformación física de materia prima para generar una geometría específica de manera automatizada. Esto se debe, a que estas máquinas cuentan con computadoras internas que permiten controlar el proceso de maquinado, así como el cambio de herramientas y el orden en la ejecución de los programas [13]. De tal forma, es necesario definir conceptualmente este tipo de sistemas para su mejor comprensión.

1.4.1 Definición de CNC.

Podemos definir a un CNC como un sistema de automatización programable que se basa en una serie de instrucciones codificadas que gobiernan todas las acciones de una máquina [13], haciendo que ésta desarrolle una secuencia de operaciones y movimientos previamente establecidos. En la figura 14, se muestran los componentes básicos de un CNC, los cuales son: *un programa*, que contiene toda la información de las acciones a ejecutar; la *unidad de control*, que interpreta las instrucciones y las convierte en señales para los actuadores de la máquina; y la *máquina*, que ejecuta las operaciones previstas [13].



Figura 14. Componentes básicos de un CNC.

1.4.2 Clasificación del CNC.

La clasificación de los sistemas CNC varía de acuerdo a ciertas características de los mismos, como son: el tipo de sistema de referencia que manejan, el tipo de trayectorias, la capacidad para controlar un número determinado de ejes simultáneamente, etc. [27]. Esta clasificación está descrita en la tabla 1.

De acuerdo a:	Tipos
Sistema de referencia	Fijo Flotante
Trayectorias	Punto a punto Paraxial Continuo
Control simultáneo de ejes	2D 2D y ½ 3D
Accionamiento	Hidráulico Eléctrico Neumático
Técnica de control	Lazo Abierto (LA) Lazo Cerrado (LC)

Tabla 1. Clasificación de un CNC [36].

1.4.3 Ventajas y Desventajas del CNC.

La conveniencia de utilizar un CNC sobre un equipo convencional, estriba en ciertos puntos como lo son: el tipo de producción que se tendrá con dichos sistemas (lotes medianos), complejidad de las geometrías a manufacturar, el número de operaciones a procesar, por cambios frecuentes en el diseño, requerimientos de precisión y fiabilidad, altos costos por falla en el maquinado, etc. [27]. Es por eso, que en la tabla 2 se denotan las ventajas y desventajas al usar este tipo de sistemas.

Ventajas:
- Mejor planeación de operaciones.
- Mayor flexibilidad en maquinado.
- Reducción en tiempo de programación.
- Mejor control de procesos en maquinados.
- Disminución en costos de herramientas.
- Se incrementa seguridad en usuario.
- Reduce el tiempo de flujo de material.
- Reduce el manejo de la pieza de trabajo.
- Aumento en la productividad.
- Se incrementa la precisión.
Desventajas:
- Alto costo de inversión.
- Personal especializado en programación.
- Alto costo en mantenimiento.
- Requiere tiempo de adaptación.
- Se requiere un volumen alto de producción.

Tabla 2. Ventajas y desventajas de un CNC [36].

1.4.4 Programación y códigos.

La manufactura de una determinada pieza sigue un proceso el cual es representado en el diagrama a bloques de la figura 15. Como primer paso es necesario elaborar un diseño de la geometría a manufacturar mediante el empleo de un determinado *software*. Enseguida, y de acuerdo con la naturaleza de la pieza, se selecciona la MH a emplear y con base en esto, podremos definir las herramientas adecuadas para cada una de las tareas así como la secuencia de uso de las mismas.

Enseguida se establecen las secuencias de corte, para después obtener determinados parámetros como coordenadas, velocidades, avances y profundidades de corte. Una vez teniendo éstos, es posible generar un programa de cada una de las operaciones por medio de un código propio de manufactura, con el fin de llevar a cabo una simulación del proceso a desarrollar.

El objetivo de una simulación, es detectar posibles malfuncionamientos en un proceso, y de ser así, realizar las modificaciones pertinentes en el diseño propuesto. Por último, y asegurándose de que la simulación sea la correcta se procede a ejecutar el programa.

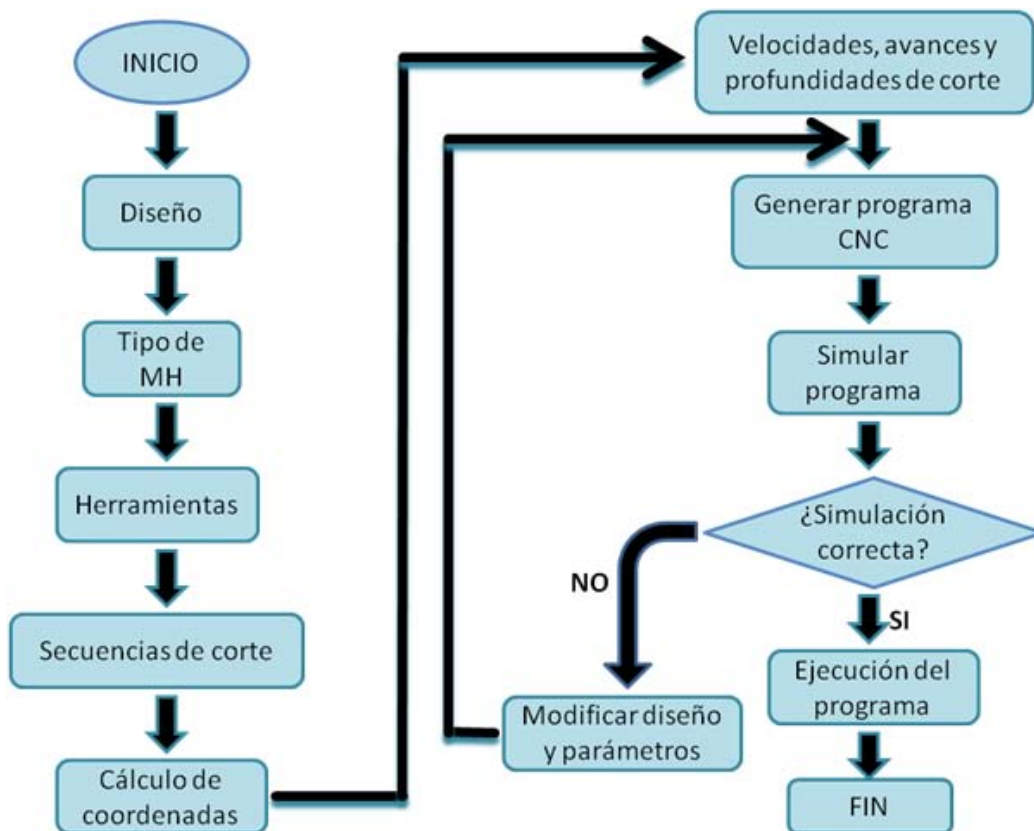


Figura 15. Proceso de manufactura en un CNC [37].

Por último, la programación en un sistema CNC está integrada por una serie de funciones codificadas (un código estándar), que permite controlar el movimiento de la herramienta en la pieza de trabajo. Ese código estándar es denominado código G y M, en donde los códigos G, se definen como *comandos preparatorios* que tienen que ver con el movimiento de los ejes de la máquina (movimientos rápidos, avances, pausas, etc.). Mientras que los códigos M, son comandos de ejecución auxiliares (*misceláneos*) que se requieren para el proceso de maquinado de las piezas, pero no son propiamente considerados como movimientos de la máquina (arranque, paro de husillo, refrigerante, cambio de herramienta, paro de programa, etc.) [13].

Una vez descrito los antecedentes básicos, se procederá a plantear la problemática, objetivo y alcances de este trabajo.



Objetivo y alcances

Actualmente, la modularidad de los diseños mecánicos permite que un mismo controlador pueda adaptarse a distintas máquinas: tornos, fresas, taladros, etc., y esto es debido a que la microelectrónica y su programación permiten incorporar nuevas funciones. Desde el punto de vista de la productividad, el hecho más relevante es que los sistemas CNC permiten un mejor uso de las máquinas por muchos factores, como son: simplificación y mejora en los métodos de programación, verificación y puesta a punto, una lectura continua del medio de almacenamiento, autocorrección en las desviaciones del proceso de maquinado, monitoreo, diagnóstico y mantenimiento de la máquina [13].

Es por esto, que muchas de las funciones de los controladores convencionales con CN han sido reemplazadas y mejoradas por los sistemas CNC con microcomputadoras integradas, donde naturalmente, el proceso de la información, el cálculo de las velocidades y las interpolaciones entre dos puntos, son realizadas por el programa de control [14].

2.1 Definición del problema.

Un diagrama detallado de un sistema CNC es mostrado en la figura 16. En éste podemos apreciar primeramente un medio de transferencia de información, el cual es conectado directamente al sistema. Éste, almacena en un bloque de memoria central el programa a ejecutar, para después ser leído y procesado por una rutina del programa de control, generando las señales requeridas para la comparación con una señal de corrección, y como resultado se generen las señales requeridas por los actuadores de la máquina en cuestión.

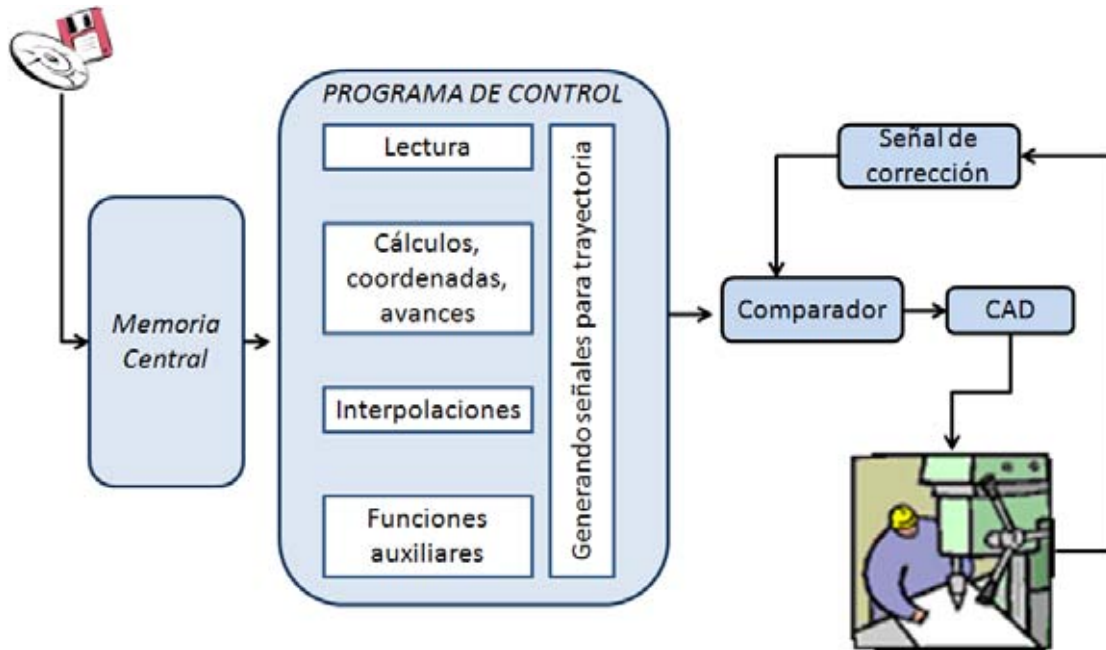


Figura 16. Diagrama general de un sistema CNC.

Analizando los beneficios proporcionados por este tipo de sistemas, el GMM, se ha inclinado por implementar (gradualmente) características propias de sistemas CNC a sus sistemas de control, con el fin de simplificar y mejorar sus métodos de programación para la manufactura de micropiezas.

En consecuencia, en el trabajo *Desarrollo de una interfaz computacional para el control de MMH* [12], aún cuando cumple con el propósito de interpretar código estándar *G* y *M* logrando así reducir considerablemente el tiempo de manufactura de piezas, sigue presentando algunos inconvenientes, como por ejemplo, las interpolaciones lineales y circulares se desarrollan punto a punto, lo cual es ineficiente y causa un incremento en el código de programa. Así mismo, la aplicación se ejecuta en ambiente *Windows*, lo que limita las frecuencias de accionamiento de los actuadores, y hace que el sistema de control sea propenso a fallas por interrupciones del sistema operativo.

Otra desventaja notoria, es que se sigue requiriendo de dos puertos paralelos en la PC para llevar a cabo el control de una MMH con cuatro grados de libertad [15]. Además, el seguimiento que hace la interfaz al proceso de manufactura, grafica una vez terminada la trayectoria, es decir, la geometría a desarrollar se representa mediante curvas ideales.

Por lo tanto, y debido a estos inconvenientes, en este trabajo se plantea una propuesta para tratar de solucionar estos problemas en la generación de microequipo del GMM.

2.2 Objetivo.

Analizar y evaluar algoritmos de interpolación aplicados al desarrollo de un sistema de control para una MMH de primera generación basado en código estándar para CNC.

2.3 Alcances.

- Establecer los antecedentes de la Micromecánica a nivel mundial y en México.
- Definir los conceptos básicos y características de los sistemas CNC (elementos, clasificaciones, tipos de códigos, programación, etc.).
- Plantear la necesidad de utilizar código estándar para CNC en las MMH del GMM.
- Recopilar información sobre algoritmos de interpolación empleados en sistemas CNC.
- Comparar y seleccionar, con base en simulaciones, el algoritmo de interpolación a emplear.
- Programar una interfaz de usuario para operar la MMH.
- Proponer, comparar y seleccionar dispositivos electrónicos para desarrollar un sistema de control basado en código estándar para CNC.
- Realizar pruebas de procesos de manufactura, empezando con piezas cuyos maquinados se efectúen mediante el movimiento de un eje a la vez, determinando así el juego mecánico.
- Realizar pruebas de procesos de manufactura que involucren el movimiento simultáneo en dos grados de libertad, y el procesamiento de programas CNC.
- Llevar a cabo las mediciones de los maquinados obtenidos, y compararlos con los maquinados desarrollados anteriormente.
- Concluir con base en los resultados obtenidos y plantear trabajo a futuro.

En el siguiente capítulo, se plantea un marco teórico acerca de las técnicas de control y algoritmos de interpolación empleados en sistemas CNC.



Marco teórico.

En un controlador CNC, tal como se muestra en la figura 17, se pueden distinguir tres bloques fundamentales: un *programa de la pieza*, un *programa de servicio*, y un *programa de control*. El *programa de la pieza* contiene la descripción de la geometría de la pieza a manufacturar y las condiciones de corte, tales como la velocidad del husillo, avance de la herramienta, profundidad de corte, etc. El *programa de servicio* es usado para verificar, editar y corregir el programa de la pieza (en caso de ser necesario). Por último, el *programa de control* interpreta las instrucciones del programa de la pieza con el fin de producir las señales requeridas por los actuadores empleados [16].

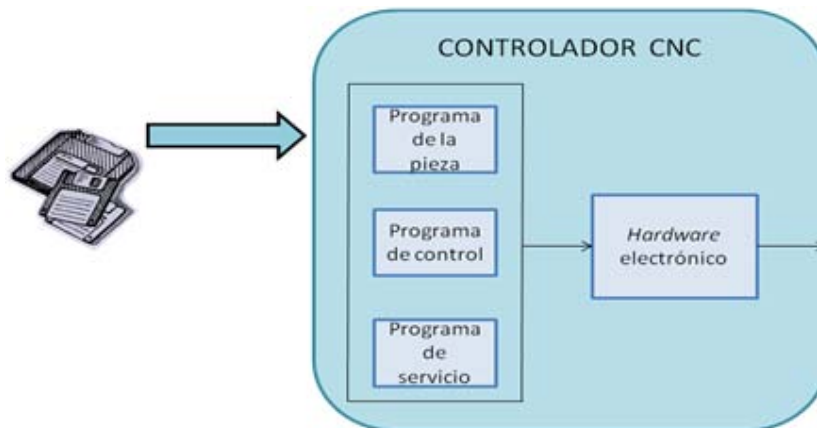


Figura 17. Bloques fundamentales de un controlador CNC.

3.1 Introducción a la programación en sistemas CNC.

Para realizar un programa en código estándar para CNC debemos tener en cuenta varios factores. Estos factores los podemos dividir en geométricos y tecnológicos. Los factores de geometría de la pieza contienen datos sobre sus dimensiones (plano de taller); además de tolerancias, acabados, superficies de referencia, origen de movimientos, etc. Mientras que los factores tecnológicos hacen referencia al material de la pieza a manufacturar, tipo de mecanizado, velocidad y profundidad de corte, revoluciones de la pieza o herramienta, lubricante, etc. [30].

Una vez atendidos estos factores se puede dar inicio a la programación de la pieza a manufacturar. Un programa de código estándar para CNC es una lista secuencial de instrucciones de maquinado que son ejecutadas por una máquina CNC [27]. A las instrucciones se les conoce como código de CNC, las cuales deben contener toda la información requerida para lograr el maquinado de la pieza.

3.2 Programación en sistemas CNC.

Se pueden utilizar dos métodos para llevar a cabo la programación en sistemas CNC, la programación manual y la programación automática. En la programación manual, el programa se escribe únicamente por medio de razonamientos y cálculos que realiza un operario. Mientras que en la programación automática, los cálculos los realiza un equipo de cómputo que suministra el programa de la pieza en lenguaje máquina, es por eso que recibe también el nombre de programación asistida por computadora [13].

3.2.1 Estructura de un programa para CNC.

En un programa para CNC cada movimiento o acción se realiza secuencialmente, por esta razón, en cada bloque debe haber ciertas características, por ejemplo, cada movimiento o acción se realiza secuencialmente, por lo que cada bloque debe ser enumerado y usualmente contener un solo comando. Estos comandos pueden ser códigos G ó M. La mayoría de los códigos G y M contienen variables (direcciones) definidas por el programador para cada función específica [37]. En la tabla 3, se describen cada una de las posibles variables que pueden acompañar, a un código G o a un código M.

La estructura de una línea de código para CNC está compuesta básicamente por cuatro elementos: una *Dirección* (N01) seguida de un *Comando* (G ó M) y después una *Palabra* compuesta de coordenadas (X10.0 Y5 Z0) para terminar con *Parámetros de manufactura* tales como velocidades, avances y herramientas.

Letra	Descripción
N	Número de secuencia
X	Comando del eje X
Y	Comando del eje Y
Z	Comando del eje Z
R	Radio desde el centro especificado
I	Coordenada X del centro del arco
J	Coordenada Y del centro del arco
F	Avance
S	Velocidad de husillo
T	Número de la herramienta

Tabla 3. Variables de la programación CNC [37].

3.2.2 Principales códigos de programación CNC.

Los códigos G que se refieren a movimientos elementales son: el G00 (movimiento rápido), G01 (avance lineal), G02 y G03 (avances en segmentos circulares). Mientras que los códigos M comúnmente empleados son: M02 (fin del programa), M03 y M04 (rotación del husillo), M05 (paro del husillo) y M06 (cambio de herramienta) [13].

Existen más códigos G y M relacionados con otras tareas, por ejemplo, la selección del plano de trabajo, pausas, sistema de unidades, retorno a posición de origen, compensaciones, sistema coordinado, activación de refrigerante, etc.

A su vez, se tienen códigos de tareas especializadas que resultan de combinaciones con los códigos elementales, a éstos se les denomina ciclos enlatados y entre ellos se encuentran: el G71 (ciclo de desbaste exterior), G72 (ciclo de desbaste y acabado vertical), G74 (ciclo de barrenado), G81, G82 y G83 (ciclos de taladrado profundo y con pausa), G92 (ciclo para hacer cuerdas), G94 (ciclo para hacer careado), etc. [27] Todos éstos representan ciclos que simplifican la programación de una pieza, combinando una gran cantidad de operaciones y disminuyendo los pasos de programación simplificando las operaciones matemáticas y optimizando las condiciones de corte [36].

Como se ha mencionado al principio de este capítulo el *programa de control* es el encargado de interpretar todos estos códigos para producir las señales adecuadas que activarán los actuadores de la máquina en cuestión. Es por esto necesario investigar más acerca de las técnicas de control empleadas en sistemas CNC, así como el tipo de algoritmos implementados en estos sistemas.

3.3 Técnicas de control en sistemas CNC.

De todos los bloques de un sistema CNC, la rutina principal en el programa de control es el algoritmo de interpolación, el cual coordina el movimiento de los ejes de la máquina en cuestión, para generar el maquinado de la ruta requerida [13]. Es importante mencionar que dicha ruta es una combinación de segmentos lineales y circulares, y de acuerdo con esto, el programa de control debe contener rutinas capaces de producir movimientos simultáneos en los ejes para desarrollar este tipo de trayectorias.

Básicamente, un sistema CNC tiene dos técnicas para transmitir las señales de referencia generadas por el programa de control, una es basada en una secuencia de pulsos (técnica por referencia de pulsos) y la otra es mediante palabras binarias (técnica por muestreo de datos) [16].

3.3.1 Técnica por Referencia de pulsos.

En los sistemas que utilizan este tipo de técnica, la computadora produce una secuencia de pulsos, como señal de referencia, para el movimiento de cada eje, cada pulso generado produce un movimiento equivalente a la resolución de la MMH. El número acumulado de pulsos representa la posición, mientras que la frecuencia de los mismos es proporcional a la velocidad en cada eje [17, 18] (figura 18).

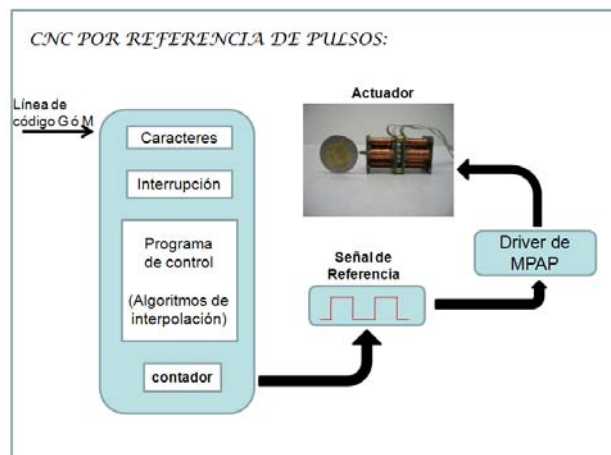


Figura 18. Sistema por referencia de pulsos.

3.3.2 Técnica por Muestreo de datos.

En este tipo de técnica (figura 19), el lazo de control es cerrado mediante *software* por medio de la misma computadora. El programa de control compara la palabra binaria generada con la señal realimentada produciendo una señal de error. Este error es enviado a un convertidor digital-analógico, el cual a su vez proporciona un nivel de voltaje proporcional requerido para la velocidad en cada uno de los ejes [19].

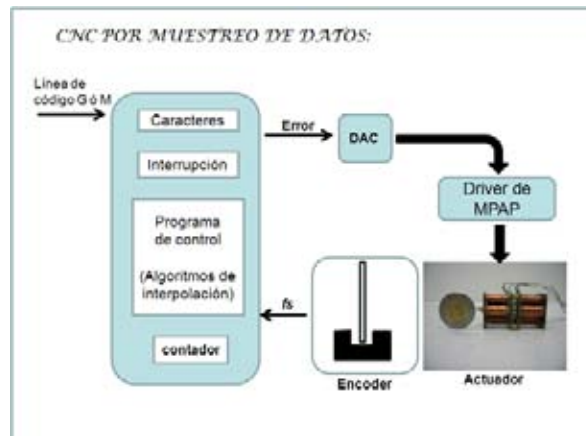


Figura 19. Sistema por referencia de palabra.

Ambas técnicas, requieren distintas rutinas de interpolación en el programa de control para generar sus correspondientes señales de referencia. En todos los sistemas de control hasta el momento desarrollados en el GMM, se ha optado por la *técnica de referencia de pulsos*, debido a la simplicidad para ser programados, y su fácil implementación para el tipo de actuadores empleados (motores a pasos). Sin embargo, hay que tener en cuenta que en este tipo de técnica hay restricción en cuanto a la máxima velocidad alcanzada durante la interpolación [17].

Los métodos de interpolación empleados en la técnica por referencia de pulsos, están basados en iteraciones controladas por la ejecución de una interrupción, es decir, cada vez que se ejecuta ésta se produce una iteración del método de interpolación, generando un pulso como salida hacia el(los) actuador(es) correspondiente(s) [20]. La máxima velocidad será proporcional a la velocidad a la cual se puede ejecutar el algoritmo de interpolación. Los algoritmos de interpolación por referencia de pulsos comúnmente empleados en sistemas CNC son básicamente tres:

1. Interpolación por Analizador Diferencial Digital, de sus siglas en inglés (*DDA-Digital Differential Analyzer*).
2. Interpolación por escalones ó incremental.
3. Interpolación por búsqueda directa, de sus siglas en inglés (*DSM-Direct Search Method*).

3.4 Interpolación por DDA.

3.4.1 Interpolación Lineal.

Este método de interpolación, está basado en adiciones, subtracciones y comparaciones, y consiste en la obtención de las principales componentes de una trayectoria rectilínea (X y Y). Suponiendo dos ejes coordenados, los valores absolutos de estas componentes, denotadas con a y b en el diagrama de flujo mostrado en la figura 20, se almacenan en los registros correspondientes de acuerdo a la proyección del eje coordenado, P_x y P_y [38].

Durante cada iteración, los valores de las proyecciones almacenados en P_x y P_y serán sumados al contenido de otro registro, denotado Q . El desborde de cada uno de estos registros (uno por cada eje coordenado, Q_x y Q_y) consiste en sobrepasar la magnitud de la hipotenusa L (condición de desborde). Además de esto, se verificarán los valores de los contadores de pasos, $CntX$ y $CntY$, cuyo valor de desborde está determinado por el cociente de las proyecciones entre la resolución del sistema.

Cuando el desborde de los registros Q_x y Q_y ocurra, un pulso es generado en la dirección correspondiente donde se originó el desborde. Una vez generado el pulso, la condición de desborde L es restada al registro suma Q donde se generó el desborde. De esta forma se actualizará el registro contador de pasos correspondiente, $CntX$ ó $CntY$. El código correspondiente a este algoritmo está en la sección A3 de los Anexos.

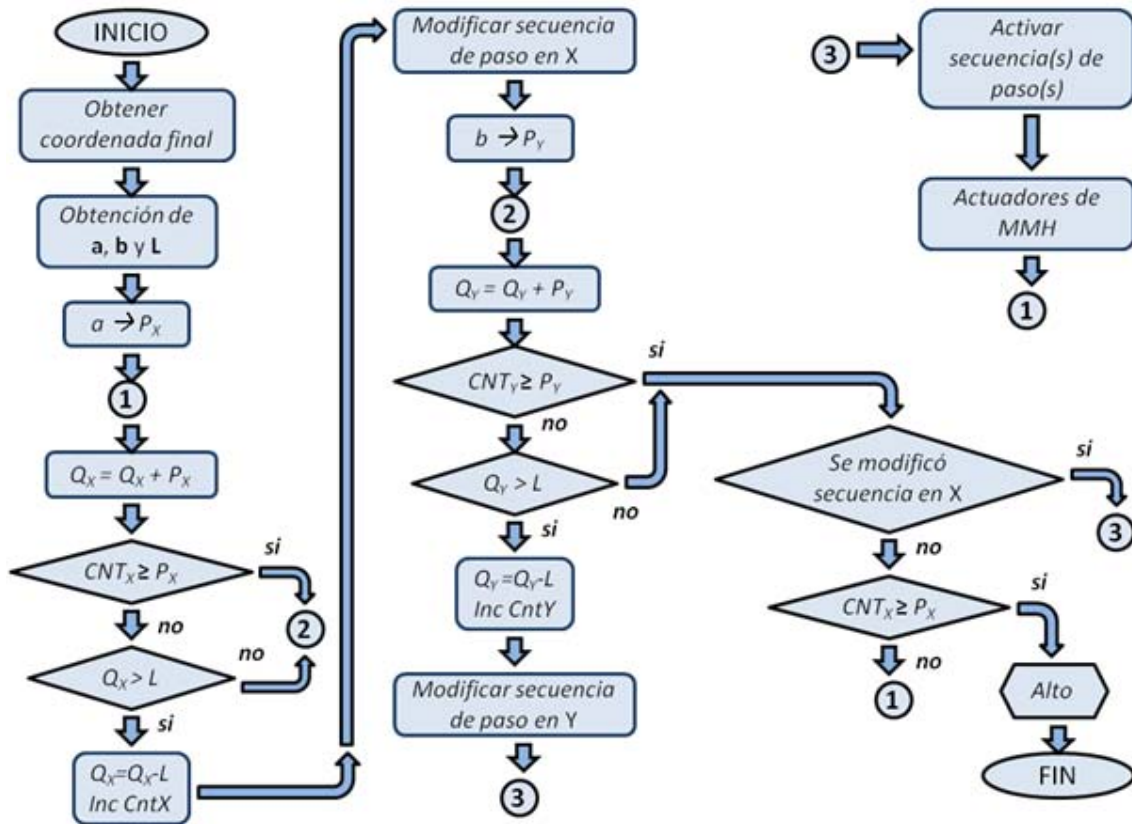


Figura 20. Algoritmo DDA para segmentos rectilíneos.

La frecuencia a la cual se ejecuta el algoritmo está dado por:

$$f = \frac{d * F * 0.1}{60} \dots\dots\dots (3.1)$$

donde el significado de las variables es el siguiente:

d : es el número de pasos necesarios para recorrer un milímetro.

F : es el avance de la herramienta durante el maquinado.

La forma empleado para obtener d será:

$$d = \frac{1mm}{Resol} \dots\dots\dots (3.2)$$

mientras, que la condición de desborde está dada por:

$$L = \sqrt{a^2 + b^2} \dots\dots\dots (3.3)$$

Este algoritmo también se puede aplicar a segmentos circulares, la diferencia existente respecto al caso de segmentos rectilíneos radica en que los valores de los registros P_x y P_y varían constantemente.

3.4.2 Interpolación Circular.

Para este caso, el método requiere de dos operaciones más en cada iteración (una adición y una substracción). Sin embargo, la metodología es muy parecida al caso lineal. La principal diferencia radica, en que al ser constante la condición de desborde (dada por el radio) las proyecciones de los registros P_x y P_y cambian al ir avanzando sobre la circunferencia [38]. En la figura 21 se muestra un diagrama a bloques de este procedimiento, y en la sección A4 de los Anexos el correspondiente código.

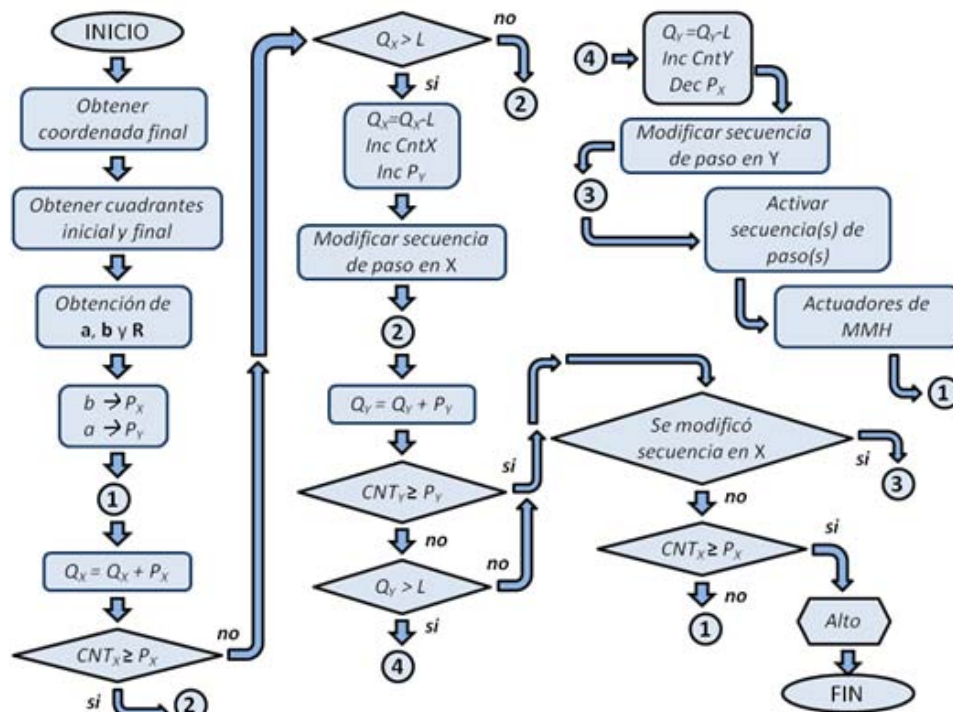


Figura 21. Algoritmo DDA para segmentos circulares (1er cuadrante).

Otras diferencias no menos importantes son: que las proyecciones X y Y (suponiendo dos ejes coordenados) formadas por los extremos del segmento circular con el centro del mismo, se almacenan en registros contrarios (X en P_y y Y en P_x). Asimismo, dependiendo del cuadrante en el que nos encontremos, al generarse un pulso en la dirección del eje X (mediante el mismo procedimiento que en el caso lineal) disminuye en una unidad el registro P_y , y si fuese un pulso para el eje Y aumenta en una unidad el registro P_x .

3.5 Interpolación Incremental (ó por escalones).

Este método, también denominado *aproximación por escalones*, realiza una aproximación a una trayectoria circular mediante incrementos (pasos) en un solo eje a la vez. En cada iteración se avanza en un solo sentido hasta alcanzar el punto final. Esto se logra mediante una variable de comparación D , dada por:

$$D_f = X_f^2 + Y_f^2 - R^2 \dots\dots\dots (3.4)$$

Como se aprecia, en la expresión (3.4) se requiere de un multiplicador que podría aumentar el tamaño de la palabra y la velocidad de procesamiento del algoritmo. En consecuencia, existe una mejora a este método [21, 22, 23] que consiste en modificar y añadir un índice más a la expresión anterior, resultando:

$$D_{i,j} = X_i^2 + Y_j^2 - R^2 \dots\dots\dots (3.5)$$

Donde i y j representan los pasos completados en cada una de las direcciones, X y Y . Ahora bien, para el caso del primer cuadrante cuando se ejecuta un paso en una determinada dirección (por ejemplo en X) la expresión (3.4) queda de la siguiente manera:

$$D_{i+1,j} = (X_i + 1)^2 + Y_j^2 - R^2 = D_{i,j} + 2X_i + 1 \dots\dots (3.6)$$

Estableciéndose un incremento representado por:

$$\Delta X_i = 2X_i + 1 \dots\dots\dots (3.7)$$

Y reemplazando (3.7) en la expresión (3.6), resulta:

$$D_{i+1,j} = D_{i,j} + \Delta X_i \dots\dots\dots (3.8)$$

Así, para el siguiente paso en X se tiene que:

$$D_{i+2,j} = D_{i+1,j} + \Delta X_{i+1} \dots\dots\dots (3.9)$$

donde:

$$\Delta X_{i+1} = 2X_{i+1} + 1 = 2(X_i + 1) + 1 = \Delta X_i + 2 \quad \dots (3.10)$$

y similarmente para el eje Y, tenemos:

$$D_{i,j+1} = D_{i,j} + \Delta Y_j \quad \dots (3.11)$$

y para:

$$\Delta Y_{j+1} = \Delta Y_j + 2 \quad \dots (3.12)$$

donde:

$$\Delta Y_j = -2Y_j + 1 \quad \dots (3.13)$$

Con esto, se mejora la velocidad de procesamiento al tener expresiones más sencillas para el valor de D (expresiones 3.8 y 3.11), en las cuales no intervienen potencias.

En la figura 22, se puede apreciar que la dirección en la que se dé el paso dependerá del signo de la variable D y del cuadrante de la circunferencia en el que se encuentre la herramienta. Así pues, si se trata del primer cuadrante y $D < 0$ (el punto se encuentra dentro del círculo) el incremento será en la dirección $X+$ (positiva), y si $D > 0$ (el punto se encuentra fuera del círculo) el incremento será en la dirección $Y-$ (negativa). El caso $D=0$ (el punto está sobre la circunferencia) puede ser incluido en cualquiera de los dos casos anteriores, y todo esto hasta alcanzar el punto final (X_f, Y_f) .

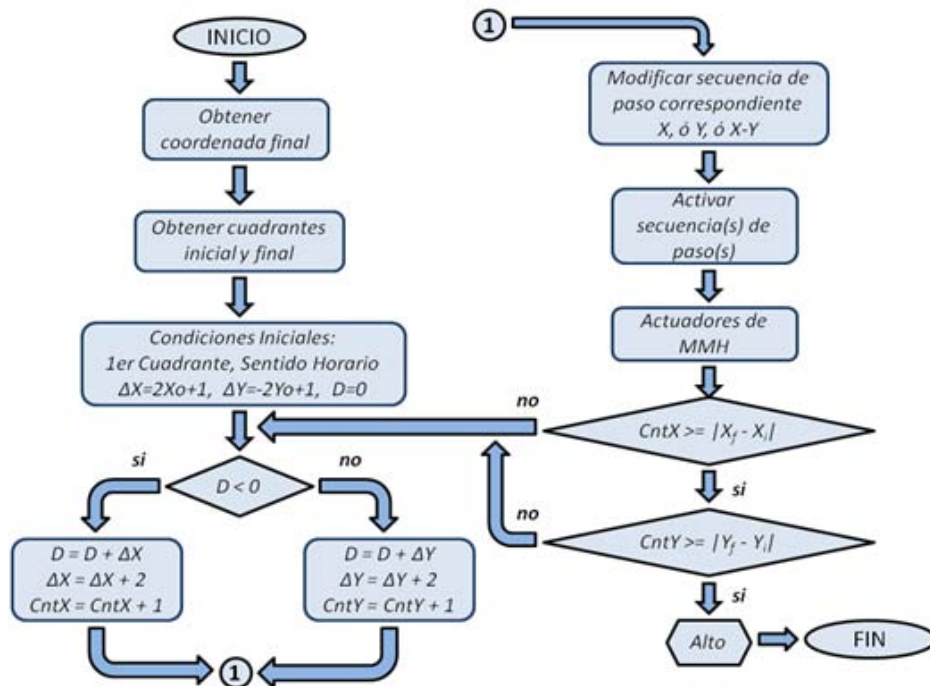


Figura 22. Algoritmo por interpolación incremental.

Por último, cabe aclarar que el valor de D que se obtiene en cada iteración, es evaluado con ayuda de su valor previo y los respectivos incrementos, ΔX y ΔY , los cuales son calculados mediante las ecuaciones (3.7) y (3.13).

La simplicidad de cada una de las ecuaciones anteriores (en este método) provoca que la velocidad de ejecución del algoritmo aumente. Sin embargo, la incapacidad de este método para originar movimientos simultáneos, provoca que más iteraciones (de las realizadas por el método DDA) sean requeridas para describir un círculo. El código correspondiente a este algoritmo es mostrado en la sección *A5 de los Anexos*.

3.6 Interpolación por Búsqueda Directa.

Este método aplica la misma técnica en los cálculos que el método de *interpolación por escalones*, con la diferencia que este método de interpolación si permite el movimiento simultáneo en ambos ejes, lo cual trae como resultado una mejor interpolación del segmento circular [20].

Esta mejora es alcanzada mediante la evaluación de la ecuación (3.5) en tres posibles puntos. Estos resultados de la variable D son comparados entre sí, con la finalidad de obtener la dirección del error mínimo. La elección de estos puntos depende del sentido de giro y del cuadrante de la circunferencia al que esté referida la trayectoria.

Por ejemplo, para un caso de primer cuadrante y sentido horario, se llevan a cabo tres pruebas para determinar la ruta del error mínimo. En la primera, una unidad (resolución) es sumada al subíndice i , teniendo que evaluar las ecuaciones (3.7) y (3.8) para obtener la correspondiente $D_{(i+1,j)}$. En la segunda prueba, se hace lo propio con el subíndice j , obteniendo de las ecuaciones (3.11) y (3.13) el valor $D_{(i,j+1)}$, mientras que en la última prueba se aplican los dos procedimientos mediante la siguiente expresión:

$$D_{i+1,j+1} = D_{i,j} + \Delta X_i + \Delta Y_j \quad \dots\dots\dots (3.14)$$

Con la ecuación (3.14) es posible obtener el tercer valor $D_{(i+1,j+1)}$ y comenzar la comparación para obtener la ruta del error mínimo. El procedimiento, se ilustra en el diagrama de la figura 23. Una vez hecha la comparación de estos tres valores se procede a generar el pulso correspondiente en la dirección resultante, además de actualizar los correspondientes valores de ΔX y ΔY , tal como se hace en el método por escalones. El código correspondiente a este algoritmo es mostrado en la sección *A6 de los Anexos*.

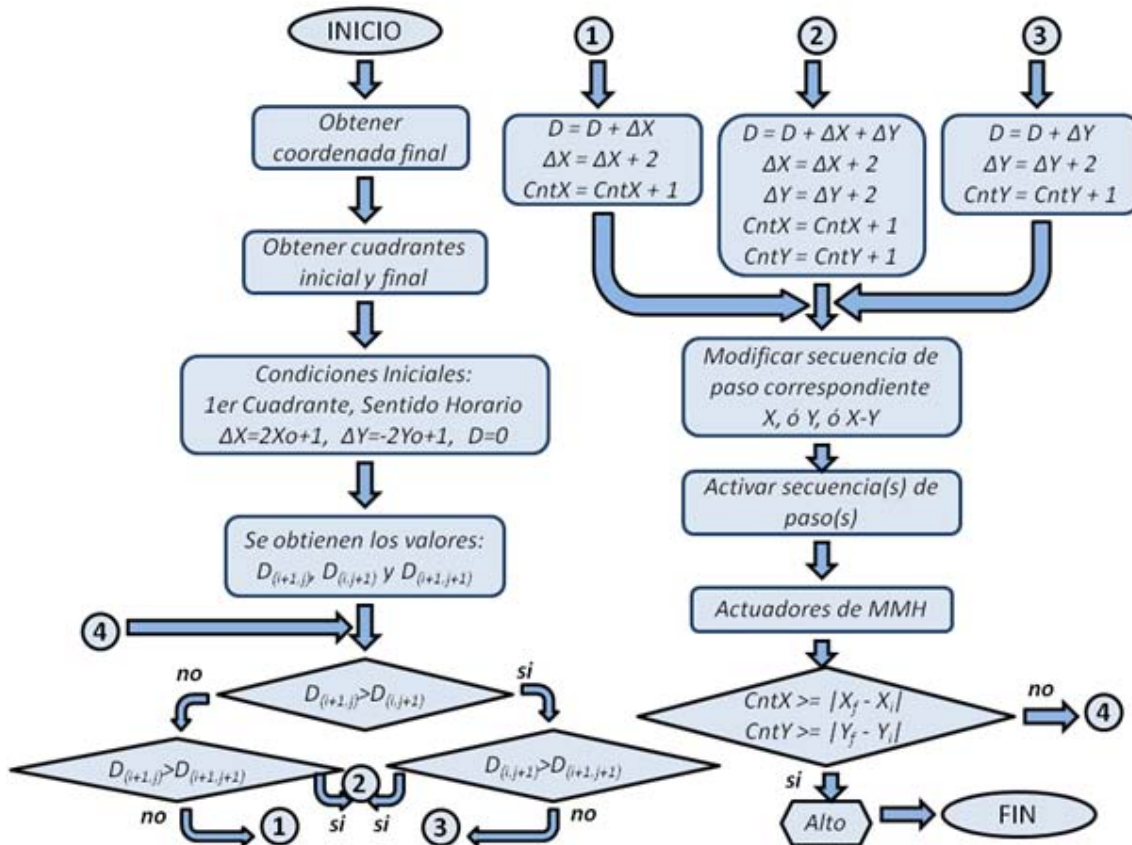


Figura 23. Algoritmo por el método de Búsqueda Directa.

Una vez planteado este marco teórico, en el siguiente capítulo, se estudiará el prototipo a controlar, se planteará una propuesta de control, y se llevarán a cabo métodos de comparación y selección para identificar las herramientas a utilizar en este proyecto. Además, se proponen dos cadenas de control para explicar el funcionamiento del programa de control.



Desarrollo del sistema de control.

El GMM, con el objetivo de mejorar el nivel de automatización de sus prototipos micromecánicos, busca que las especificaciones en el desarrollo de sus sistemas de control sean compatibles a los sistemas CNC comerciales. Sin embargo, antes de adentrarnos en las características que deberá tener este sistema de control, describiremos primeramente el prototipo de MMH a controlar, haciendo énfasis en su estado actual y principales componentes.

4.1 Caso de estudio.

Actualmente, se cuenta con una nueva versión del primer prototipo de MMH mostrado en la figura 5. En esta nueva versión se aplicaron sistemas CAD y CAM con la finalidad de mejorar la reproducibilidad del prototipo, además de facilitar la manufactura y ensamble del centro de maquinado mostrado en la figura 24.

Esta nueva versión de MMH tiene dimensiones de 120x160x85mm, tres ejes traslacionales y ortogonales, y uno rotacional. La transmisión de engranes tienen una relación de 1:128, mientras que sus actuadores son motores a pasos, con 28 pasos por revolución (figura 25). El desarrollo de estos actuadores es detallado en el trabajo de tesis *Desarrollo de motores de pasos para micromáquina herramienta* [28]. La resolución de este prototipo es de 140[nm] por paso del motor. El desarrollo detallado de este prototipo se encuentra registrado en el trabajo de tesis *Aplicación de sistemas CAD y CAM para el desarrollo de una micromáquina herramienta* [6].



Figura 24. Nueva versión del primer prototipo de MMH desarrollado en México.

El diseño de este prototipo está pensado para implementar un sistema de control mediante una PC. En consecuencia, se espera que el sistema de control sea capaz de reconocer código estándar para CNC, con el fin de generar las señales de control adecuadas para una aproximación óptima de trayectorias lineales y circulares de las diferentes geometrías a manufacturar.

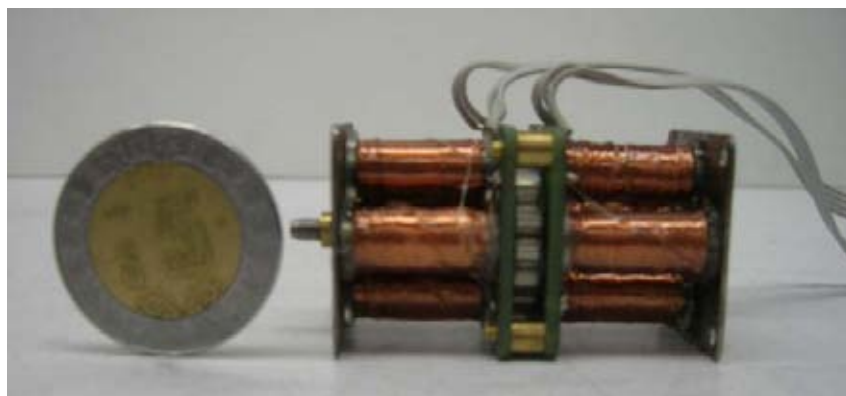


Figura 25. Motor a pasos bipolar de magneto permanente.

Es momento de establecer las características con las que deberá contar esta propuesta de control.

4.2 Características esperadas.

- El sistema de control propuesto debe de reconocer las principales instrucciones G y M para procesos de fresado descritas posteriormente en el apartado 4.3.2.
- Los algoritmos de control deben ser programados en un dispositivo externo a la PC con el fin de reducir la dependencia de la misma.
- El algoritmo de interpolación programado debe ser aquel que ofrezca los mejores parámetros en su ejecución (aproximación, iteraciones, complejidad, etc.).
- La interpretación gráfica (en la interfaz) del proceso de manufactura debe monitorearse en cada paso del motor.
- El sistema de potencia debe ser capaz de proporcionar las características eléctricas necesarias para el accionamiento de los actuadores.
- La interfaz de usuario programada debe monitorear, en sus diferentes modos de operación, parámetros básicos de manufactura (avance, revoluciones por minuto, posición y proceso de maquinado).

4.3 Propuesta de control.

Como ejemplo de un sistema de control para un MH convencional, el GMM cuenta con una fresadora CNC de la empresa DynaMechtronics mostrada en la figura 26. Esta MH lleva a cabo un control por procesamiento distribuido mediante Microcontroladores de T.I (*Texas Instruments*). Estos dispositivos son generalmente empleados en tareas de automatización debido a su bajo costo y amplia flexibilidad para diseño e implementación (ver tabla 4). Entre otras características, esta fresadora permite ser operada por desplazamientos continuos, por rutinas de pasos establecidas, por posicionamiento a través de una perilla, por una línea de código G ó M, y por un programa para CNC.

En consecuencia, en el diagrama de la figura 27 se muestra una propuesta de control para una MMH de primera generación, que mediante algoritmos de control programados en un dispositivo externo a la PC, disminuirá la dependencia que se ha tenido con ésta en los últimos sistemas de control desarrollados.



Figura 26. Fresadora CNC del GMM [6].

Microcontroladores	
Campo de aplicación	Automatización industrial Sistemas de navegación Aparatos portátiles Robótica Electrodomésticos Sistemas de seguridad y alarmas, etc.
Tipos de memoria	ROM, OTP, EPROM, EEPROM y FLASH
Periféricos comunes	Temporizadores, Convertidores, PWM, Canales de comunicación serie y paralelo, Puerto de E/S, Interrupciones, etc.
Ventajas	Costo Flexibilidad Consumo energético Tamaño
Desventajas	Pocas instrucciones Capacidad de memoria
Compañías que los fabrican	Intel, AMD, Motorola, IBM, TI, Microchip, etc.
Gamas	Enana, Baja, Media, Alta y Mejorada.

Tabla 4. Características de los Microcontroladores.

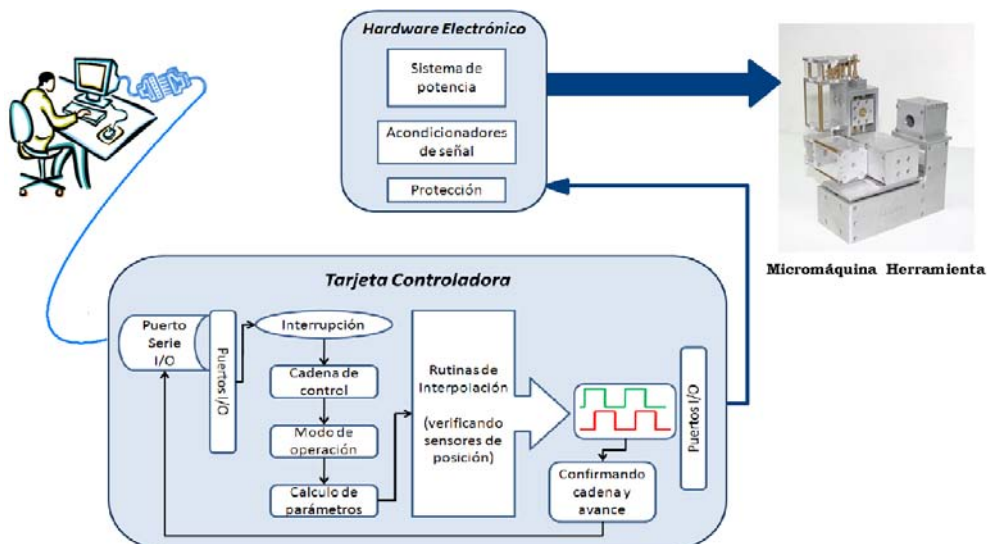


Figura 27. Sistema de control propuesto.

Tal como se muestra en el diagrama de la figura 27, la tarjeta controladora contará con ciertos procesos programados, tales como, atención a interrupción, recepción de cadenas, cálculo de parámetros, interpolaciones, etc. Todo esto con el fin de que el *hardware* electrónico (potencia) proporcione los niveles de corriente adecuados a las señales de referencia generadas (control) para el accionamiento de los actuadores.

Así pues, empleando los recursos del GMM que puedan satisfacer especificaciones tales como: gran capacidad de memoria programa, entradas y salidas para controlar más de cuatro grados de libertad, procesamiento a 20MHz (ó más), set instrucciones elaborado, cierto grado de conocimiento para su programación, etc., se implementó al sistema propuesto una tarjeta de *Microchip* con un microcontrolador PIC18F8720. Las características de esta tarjeta se muestran en la tabla 5. Mientras que los recursos del microcontrolador PIC se resumen en la tabla 6.

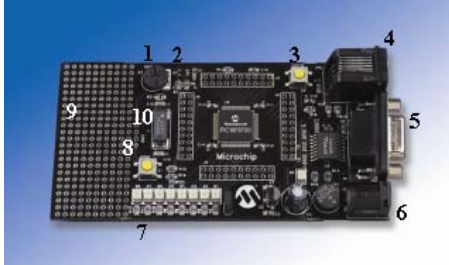
Tarjeta electrónica empleada	
	
Potenciómetro conectado al CA/D de 10bits.	(1)
Sensor de temperatura: Microchip TC74.	(2)
Pulsador de <i>reset</i> .	(3)
Conector para MPLAB ICD2.	(4)
Conector RS232.	(5)
Terminal de alimentación.	(6)
Hilera de 8 <i>leds</i> , con <i>jumper</i> para desconectar.	(7)
Pulsador para modalidad de programación.	(8)
Área de propósito general.	(9)
Cristal.	(10)

Tabla 5. Características de la tarjeta de control.

4.3.1 Características del sistema de control.

En el desarrollo de este sistema de control, se abordan tres aspectos: La PC debe estar limitada a desarrollar tareas de monitoreo, en los diferentes modos de control del sistema. El programa de control debe ser capaz de calcular y ejecutar interpolaciones de tipo lineal y circular para obtener movimientos suaves y precisos. Por último, el sistema reconocerá código estándar G y M para tareas de fresado.

Microcontrolador.	PIC18F8720
Tipo de encapsulado.	80pines-TQFP
Frecuencia de operación [MHz].	20
Memoria de programa (bytes).	128K
Memoria de datos (bytes).	3840
Memoria EEPROM (bytes).	1024
Interfaz para memoria externa.	Si
Fuentes de interrupción.	18
Puertos de entrada/salida.	A, B, C, D, E, F, G, y J
Pines de entrada/salida.	68
Temporizadores.	5
Módulos CCP.	5
Módulos de comunicación serial.	MSSP, USART (2)
Módulos de comunicación paralela.	PSP
Módulos CAD.	10bits / 16 canales
Set de instrucciones.	77

Tabla 6. Recursos del microcontrolador PIC18F8720.

4.3.1a Modos de operación.

Se propone que el sistema de control a desarrollar cuente con cuatro modos de operación para accionar el movimiento en la MMH:

- Por Cursor de mando (*Joystick*).
- Por una Secuencia de Pasos.
- Por ejecución de una Línea de código G y M.
- Por ejecución de un programa de código G y M.

En el modo por *Joystick* la interfaz computacional a desarrollar permitirá que, mediante un cursor de mando programado, el usuario pueda desplazar los carros de la MMH en todo su espacio de trabajo, logrando ubicar la herramienta dentro del volumen de trabajo además de variar la velocidad de desplazamiento como se considere adecuado.

En el modo *Por Pasos*, los avances en la MMH serán por rutinas cuyos valores son los siguientes: 1 paso, que representa el mínimo desplazamiento del sistema, 14 y 28 pasos, que representan media vuelta y una vuelta, respectivamente, 10, 50, 100, y 1000 pasos, rutinas que permiten completar desplazamientos poco críticos en el posicionamiento de la herramienta.

Los dos modos de operación restantes que ejecutan una línea de código G ó M y un programa son muy parecidos aunque con propósitos diferentes. Mientras que el modo de ejecución de una línea de código es empleado en cualquier instante de un proceso de manufactura, o bien, para establecer el cero de programa; en el modo de ejecución de programa se ejecuta consecutivamente cada una de las líneas del programa CNC dando por hecho que todas las condiciones para el proceso de maquinado están bien definidas.

4.3.1b Parámetros monitoreados.

Durante la ejecución de cada uno de los modos de operación del sistema de control es necesario que el usuario tenga en cuenta cómo varían ciertos parámetros que influyen en el proceso de maquinado. Los parámetros que este sistema de control monitorea en todas las modalidades de funcionamiento, son:

- Posición.
- Velocidad (Hz y RPM's).
- Avance (mm/min).
- Cadena de control enviada.
- Interpretación gráfica.

La posición está representada por tres tipos de coordenadas, las que reflejan la posición respecto al cero de máquina (X_m , Y_m , y Z_m), otras que dan la posición respecto al cero de pieza (X_p , Y_p , y Z_p), y las últimas que nos dan las coordenadas del cero de pieza (X_{cp} , Y_{cp} , y Z_{cp}). Mientras tanto, la velocidad y el avance serán parámetros referidos a los actuadores y a la herramienta empleada en la MMH, y estarán restringidos de acuerdo a sus límites de operación.

En cuanto al envío de información, se confirmarán las cadenas de control enviadas entre ambas etapas, interfaz computacional y tarjeta de control, dando la posibilidad al usuario en todo momento de localizar fallas de sintaxis. Por último, la interpretación gráfica consiste en generar un gráfico representativo de lo manufacturado, y a diferencia del sistema de control anterior denominado *Desarrollo de una interfaz computacional para el control de MMH*, el avance de cada actuador se monitorea en todo instante lo que permite tener una representación gráfica paso a paso del proceso de manufactura.

4.3.2 Códigos G y M soportados.

Los sistemas de CNC se gobiernan por un conjunto de aproximadamente 70 instrucciones de código *G* y 25 instrucciones de código *M* [27]. La cantidad de instrucciones *G* y *M* pueden llegar a variar dependiendo del fabricante del sistema CNC. Para el caso del sistema propuesto se contempla que contenga las instrucciones *G* y *M* estándares para llevar a cabo tareas de fresado en dos y medio ejes, es decir, el movimiento de un punto a otro se realizará moviendo tres ejes pero sólo dos de manera simultánea. Los códigos estándares *G* y *M* empleados, más otros códigos elementales para tareas de fresado son:

1. G00 Posicionamiento rápido.
2. G01 Interpolación lineal.
3. G02 Interpolación circular en sentido horario.
4. G03 Interpolación circular en sentido anti-horario.

5. G04 Parada temporizada.
6. G28 Retorno automático a posición cero de máquina.
7. G54 Localidad para almacenar y referenciar el cero de pieza.
8. G90 Sistema coordinado absoluto.

Mientras que las funciones M estándares empleadas son:

1. M30 Fin de programa.
2. M03 Activación del husillo en sentido horario.
3. M04 Activación del husillo en sentido anti-horario.
4. M05 Paro del husillo.
5. M06 Cambio de herramienta.

4.3.2a Descripción y sintaxis de los códigos.

La descripción y sintaxis de cada una de las instrucciones del apartado anterior se muestran en las tablas 23 y 24 de la sección de *A1 de los Anexos* de este trabajo. En la mayoría de los códigos G listados en el apartado anterior se requiere el traslado de la herramienta de un punto a otro, esto hace necesario la ejecución de algún método de interpolación. En el capítulo tres se estudiaron tres métodos de interpolación empleados en sistemas CNC. En el siguiente apartado se muestran simulaciones como resultado de haber programado los tres métodos de interpolación en un *software* computacional.

4.4 Simulación de los métodos de interpolación.

Con el fin de analizar los métodos de interpolación hasta ahora expuestos, se programaron y simularon los tres métodos en un programa computacional. El *software* empleado para esta tarea fue *Visual Basic*. Aunque actualmente se manejan en el GMM tres opciones de *software* para el desarrollo de interfaces computacionales (*Borland C++*, *Visual Basic*, y *LabVIEW*), la elección fue debido a que el conocimiento que se posee de *Visual Basic* redujo el tiempo de aprendizaje y el de desarrollo de la aplicación en comparación con las otras dos opciones. En las figuras de la 28 a la 31 se puede apreciar el resultado de estas simulaciones.

La aplicación programada mostrada en la figura 28 es capaz de simular trayectorias circulares (de 50 μ m de radio máximo) empleando cualquiera de los tres métodos de interpolación hasta ahora descritos, a su vez estas simulaciones son comparadas con una trayectoria ideal. Además, esta aplicación permite dibujar las tres trayectorias empalmadas con el fin de comparar la ejecución de los tres algoritmos.

Otro parámetro que se maneja en la aplicación mostrada en la figura 28, es el número de iteraciones que lleva a cabo cada método de interpolación. Este parámetro, se emplea para determinar cuál de los métodos propuestos requiere el menor tiempo para describir la misma trayectoria.

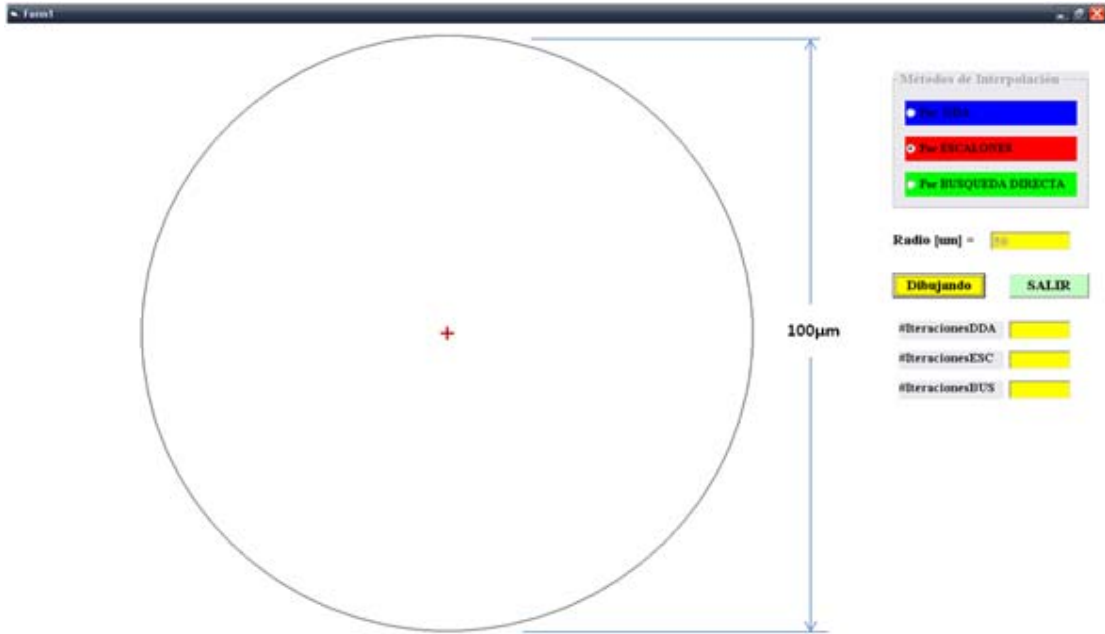


Figura 28. Pantalla del programa que simula los métodos de interpolación.

En la figura 29 se puede apreciar la simulación del método de *interpolación por escalones* para una circunferencia de $30\mu\text{m}$ de radio. En esta representación gráfica se observan la ruta generada por este algoritmo (en rojo), y el burdo de su aproximación en las zonas 1, 2, 3 y 4 de la circunferencia, es decir, en las fronteras de cada uno de los cuadrantes.

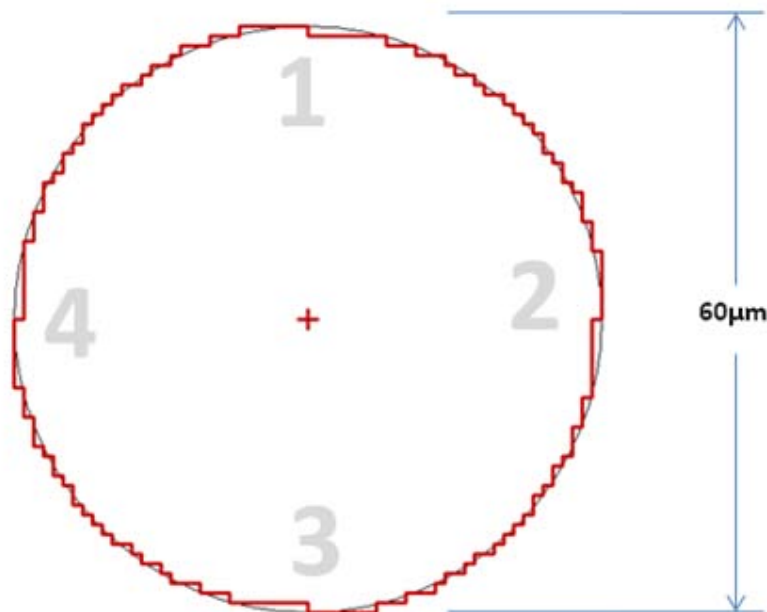


Figura 29. Simulación del método por escalones.

Por otra parte, en la figura 30 se puede observar la simulación (en azul) del método de *interpolación por DDA*, para la misma circunferencia que el método por *escalones*. A diferencia del método por *escalones*, en esta representación gráfica se aprecia una mejora en la descripción de la curva en las zonas 1, 2, 3 y 4 de la circunferencia. También, se observa en la simulación que este método tiene una mejor aproximación en el resto de la circunferencia ideal, debido a la implementación de pasos simultáneos.

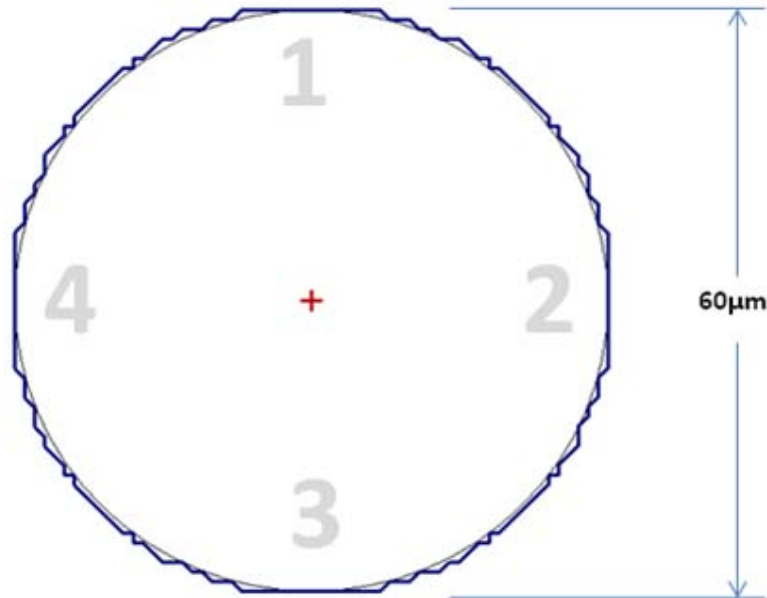


Figura 30. Simulación del método por DDA.

En la tercera simulación mostrada en la figura 31 se observa la trayectoria generada (en verde) por el método de *interpolación por búsqueda directa*, para una circunferencia de 30 micrómetros de radio. En esta representación, se aprecia como el seguimiento a la circunferencia ideal es más apegado que en los dos métodos anteriores. Los resultados de estas tres simulaciones para una circunferencia de 30 micrómetros de radio son mostrados en la tabla 7.

Por último, en la figura 32 se observa la simulación de las tres trayectorias generadas por los tres métodos de interpolación (empalmadas). En esta simulación se aprecia como el método por *búsqueda directa* (en verde) optimiza la trayectoria calculando el error generado en las tres posibles direcciones dependiendo del cuadrante. Mientras que en la figura 33 se muestran tres acercamientos correspondientes a las zonas 1, 5 y 2 de la figura 32.

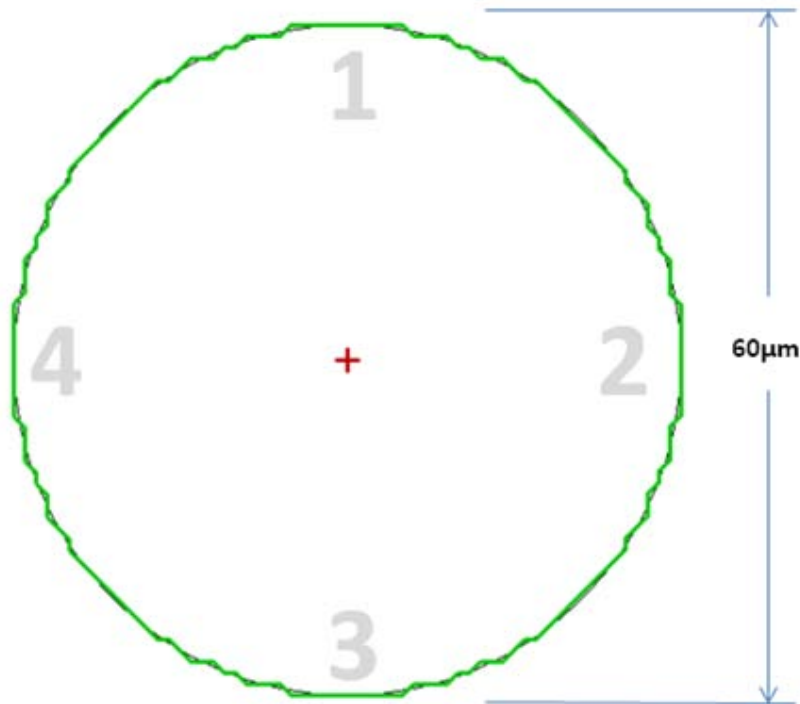


Figura 31. Simulación del método por Búsqueda Directa.

Algoritmo	Número de Iteraciones (CIRCUNFERENCIA)	Error radial teórico máximo [nm]	Tiempo por iteración [μ s]
Por escalones	240	140	3.2
Por DDA	188	140	6.8
Por búsqueda directa	168	70	8.6

Tabla 7. Resultados obtenidos de las simulaciones a los algoritmos propuestos.

Otro parámetro importante a considerar en estos algoritmos es el tiempo que tardan en ejecutar una iteración con el fin de tomar la decisión respecto a la dirección en la cual hay que avanzar.

Como se aprecia de la tabla 7, el método por escalones al no implementar pasos simultáneos requiere de menos comparaciones para determinar una ruta de avance, esto conlleva a tener el menor tiempo consumido por iteración resultando en 3.2μ s. Por otro lado, el método por búsqueda directa al ser prácticamente una mejora del método por escalones, ya que implementa pasos simultáneos, tiene un consumo de tiempo de 8.6μ s el cual es mayor al registrado por los demás. Finalmente, y debido al número de comparaciones realizadas, el método por DDA le lleva 6.8μ s el ejecutar una iteración para decidir la dirección de avance.

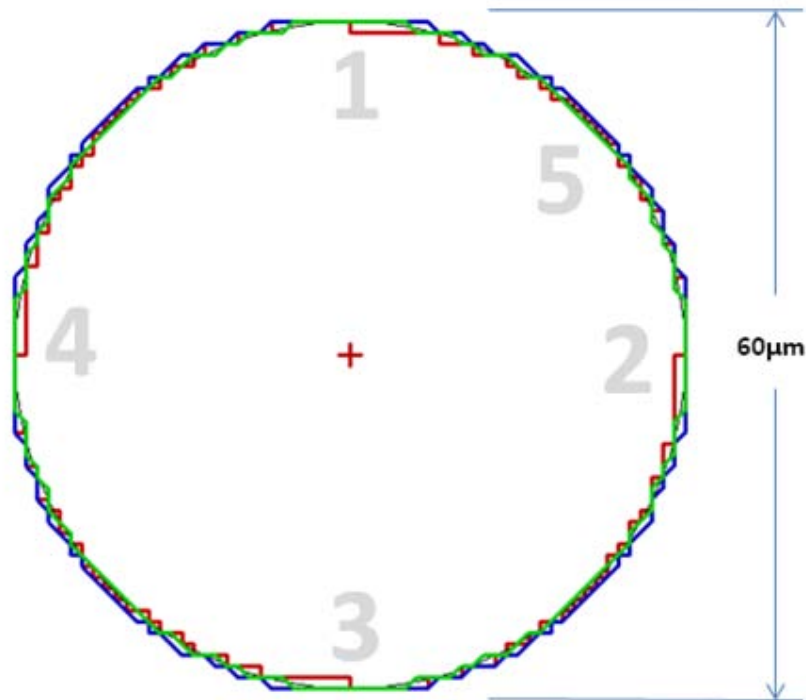


Figura 32. Simulación de los tres métodos de interpolación.

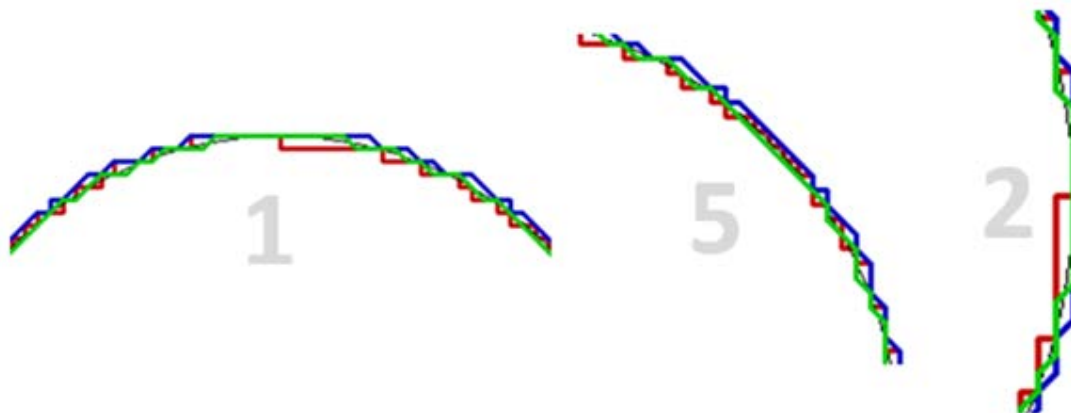


Figura 33. Acercamientos a las zonas 1, 5 y 2 de la figura 32.

4.5 Evaluación de los métodos de interpolación.

Hasta el momento se han descrito tres métodos de interpolación para sistemas CNC, y en este apartado, se evalúan con el fin de sentar una base para la selección apropiada del método de interpolación a emplear. El proceso de evaluación utilizado, denominado *Matriz de decisión*, pone especial énfasis en lo que se busca (objetivo). La aplicación de este proceso se describe en los siguientes pasos:

- I. Establecer los parámetros de importancia (objetivos, rangos ó criterios).
- II. Ordenar los parámetros de manera jerárquica.
- III. Asignar valores relativos a cada parámetro.
- IV. Establecer el rendimiento para cada solución propuesta (calificar la propuesta).
- V. Calcular y comparar en función de los resultados [1], [39].

Desarrollando los pasos I, II y III, se obtiene la tabla 8 donde se enuncian y ordenan jerárquicamente los parámetros de importancia así como los valores relativos asignados a cada parámetro. El nivel de importancia de cada valor relativo está asignado de la siguiente manera: el valor 1 significa *poco importante*, el valor 2 significa *importante*, mientras que el valor 3 significa *muy importante*. La suma de los valores relativos deberá corresponder a 10.

Parámetros de importancia	Valor relativo asignado
Mejor Simulación.	3
Error radial teórico máximo.	3
Número de iteraciones (CIRCUNFERENCIA).	3
Tiempo por iteración.	1

Tabla 8. Parámetros de importancia y valores relativos asignados.

4.6 Selección del método de interpolación.

Para establecer el rendimiento de cada solución propuesta (calificación) se emplea una escala del uno al diez. Enseguida se multiplica la calificación otorgada por el valor relativo asignado (factor de importancia) a cada parámetro, esto conlleva a obtener un valor máximo de hasta 100 puntos por cada solución propuesta. La suma más alta de estos *productos* determina la opción a escoger. Los resultados del proceso de evaluación a estos algoritmos son mostrados en la tabla 9.

El rendimiento de cada solución (calificación), de acuerdo a cada parámetro, se realizó de la siguiente manera: para el parámetro *Mejor Simulación*, se consideraron las simulaciones mostradas en las figuras de la 29 a la 32, y se estableció un calificación de 10 a la circunferencia ideal. Además, se tomó en cuenta el grado de aproximación en las zonas 1, 2, 3 y 4 de las figuras mencionadas anteriormente. En cuanto al parámetro *Error radial teórico máximo*, se determinó que la calificación de 10 correspondiera a un error cero, y que la calificación fuese disminuyendo de acuerdo a las unidades de longitud básica del sistema (resolución) que fuese incrementando el error a partir de cero.

Método de Interpolación	Mejor Simulación (3)		Error radial teórico max. (3)		Número de Iteraciones (CIRCUNFERENCIA) (3)		Tiempo por iteración [μs] (1)		TOTAL
DDA	8	24	9	27	8.8	26.4	7	7	84.4
Escalera	6	18	9	27	5.7	17.1	10	10	72.1
Búsqueda Directa	9	27	9.5	28.5	10	30	5	5	90.5

Tabla 9. Evaluación de los algoritmos de interpolación.

Mientras tanto, en el parámetro *Número de Iteraciones* se seleccionó como referencia el que menor número de iteraciones realizó (para una calificación de 10), y dicho número de iteraciones se dividió entre 100, lo cual representó la resolución para ir disminuyendo la calificación. Por último, en el parámetro de *Tiempo por iteración* se decidió seleccionar como referencia una calificación de 5 para el que mayor tiempo requirió para una iteración, y después dividir esa magnitud entre 10 para ir incrementando la calificación en las demás lecturas.

Concluyendo con base en los resultados de la tabla 9, el método de interpolación con mayor puntuación fue el de *búsqueda directa* logrando una suma de 90.5 puntos. Su principal ventaja fue la de generar una trayectoria simulada con una mejor aproximación a la curva ideal que los otros dos métodos. Aunque el tiempo de cada iteración es aproximadamente 2.7 veces el tiempo ejecutado por el método de escalera, compensa este defecto con el número de iteraciones realizadas para describir una circunferencia.

Una vez seleccionado el algoritmo de interpolación a emplear, se procede a plantear el proceso de control que permita reconocer los códigos propuestos.

4.7 Programa de control.

En la figura 34 se muestra un diagrama de flujo que hace mención a las principales rutinas del programa de control. En este diagrama se aprecia que después de las tres primeras rutinas el programa se ramifica hacia subrutinas de acuerdo a la cadena de control recibida. Cabe mencionar que todos los códigos manejados por esta aplicación, y su sintaxis, son mostrados en la sección de *A1 de los Anexos*.

En el primer bloque, *INICIO*, se declaran e inicializan todas las variables del programa, además de configurar la interrupción por recepción serial, la transmisión de datos, los puertos de entrada y salida, etc.

El bloque *RECEPCIÓN* se refiere a la rutina encargada de almacenar la cadena de control enviada por la interfaz computacional. Para asegurar que la cadena se haya recibido completa y correctamente el programa de control envía un eco a la interfaz, y ésta la despliega en una casilla dedicada a este propósito cuyo nombre es *CONFIRMACIÓN*.

Por otro lado, el bloque *MODALIDAD*, analiza el primer carácter de la cadena recibida y con base en esto genera la ruta de procesamiento de acuerdo al modo descifrado. Son 11 los posibles caminos en este programa de control:

- 1) Rutina de *alto*.
- 2) Rutina para modificar el juego mecánico (*backlash*).
- 3) Rutina para avanzar con un cursor programado (*joystick*).
- 4) Rutina para avanzar moviendo dos ó más ejes.
- 5) Rutina para avanzar con una secuencia de *pasos*.
- 6) Rutina para accionar el motor a una frecuencia determinada.
- 7) Rutina para regresa los carros al origen de máquina (*home*).
- 8) Rutinas para incrementar y disminuir la velocidad (sólo disponibles en uso por joystick).
- 9) Rutina para almacenar el cero de pieza.
- 10) Rutina para interpretar una línea de código *G*.
- 11) Rutina para interpretar una línea de código *M*.

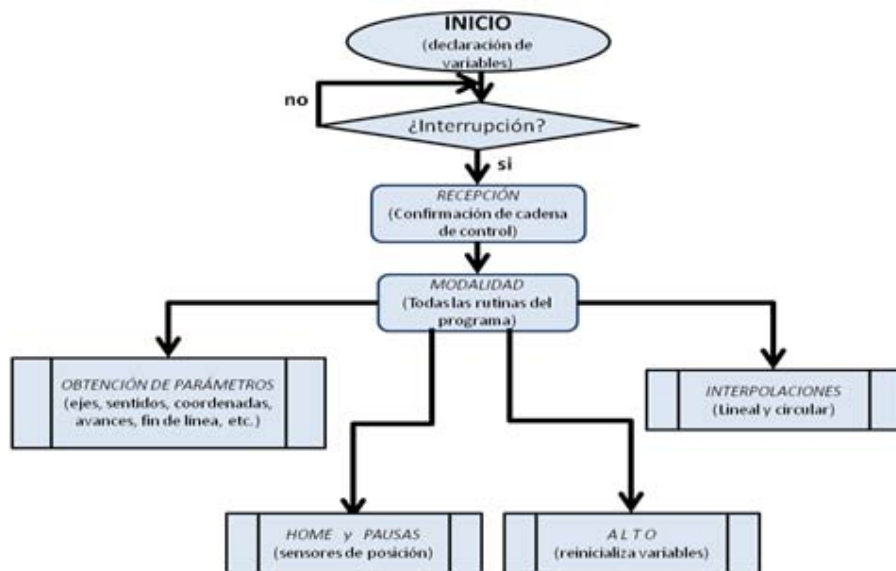


Figura 34. Rutinas y sub-rutinas principales del programa de control.

Los últimos cuatro bloques hacen referencia a todas las sub-rutinas listadas arriba. En el apartado 4.9 de este capítulo se muestran cuatro ejemplos a nivel diagrama de cómo procesa el programa de control una cadena enviada por la interfaz computacional.

4.8 Interfaz computacional.

La interfaz computacional desarrollada en *Visual Basic* consta de dos ventanas: una relativa a la presentación del proyecto llamada *pantalla de inicio* la cual se muestra en la figura 35, y otra dedicada a la operación del prototipo de MMH llamada *interfaz de usuario* la cual se muestra en la figura 36.

En la *pantalla de inicio* esencialmente se observa información relacionada con la dependencia de origen, título de proyecto, autores, país, fecha, y dos botones. El botón *INICIAR* manda a llamar la ventana de la interfaz, con la cual el usuario puede comenzar a operar la MMH. Mientras que el botón *SALIR* cierra la aplicación.



Figura 35. Pantalla de inicio de la interfaz computacional.

En la *interfaz de usuario* mostrada en la figura 36 se muestran nueve secciones. En la sección número 1, tenemos un cuadro programado referente a las modalidades en las que puede operar el prototipo: por joystick, por pasos, por línea y por programa, la selección de cualquiera de estas opciones habilitará los cuadros correspondientes a estas modalidades.

La sección número 2 se refiere a la modalidad por joystick, y en ella se aprecia una imagen del prototipo de MMH a controlar. Sobre dicha imagen, se muestran una serie de botones con los cuales se le indica al usuario de qué eje y en qué dirección se trasladará el carro, además de dos botones que permitirán incrementar ó disminuir la velocidad durante el desplazamiento. También se logra distinguir en la parte superior un botón que activa el movimiento de dos ó más ejes, simultáneamente, esto ocurrirá siempre y cuando se ponga la sintaxis adecuada (ver sección A.2 de los Anexos) dentro de la casilla próxima al botón rotulado *Activa*.

Por otro lado, la sección 3 es una modalidad dedicada a los avances que se encuentran previamente definidos por 6 secuencias de pasos (1, 14, 28, 10, 50, 100, y 1000), y se pueden ejecutar pulsando los botones que se encuentran etiquetados de la misma forma que en la sección de joystick, con el fin de que el usuario tenga conocimiento de la dirección y sentido de desplazamiento.

En la sección 4 se dispone de una casilla para que el usuario introduzca una línea de código *G* ó *M*, ó bien una cadena que active alguna de las rutinas listadas en la sección 4.3.2 de este capítulo. La cadena introducida en esta casilla es enviada carácter por carácter vía puerto serie, y de la misma forma es recibida para su confirmación y colocada en la casilla de la sección 5 con el fin de que el usuario verifique su cadena de control.

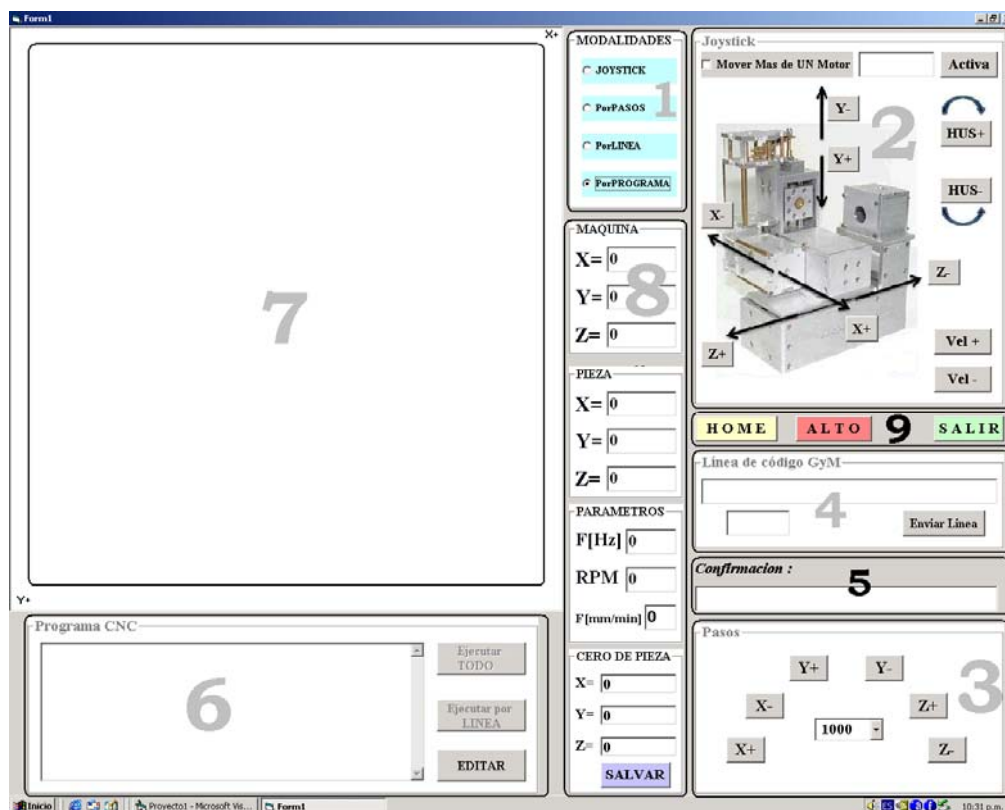


Figura 36. Secciones de la interfaz computacional.

En la sección 6 se tiene una caja de texto en la que se puede introducir un programa CNC con una debida sintaxis. Este programa puede ser editado y ejecutado línea por línea. Mientras que en el área de la sección 7 se representa gráficamente el resultado de cada una de las modalidades descritas hasta el momento. Los movimientos realizados en el modo joystick y por pasos serán graficados en color gris. Los movimientos rápidos referentes a los códigos *G00* y *G28*, además del regreso a cero de máquina, serán graficados en color amarillo. Mientras que las interpolaciones representadas por los códigos *G01*, *G02*, y *G03* serán graficadas en color azul.

La sección 8 es empleada para tareas de monitoreo. En esta sección se registran tres tipos de coordenadas: las primeras son relativas al cero de máquina (X_m, Y_m, Z_m), otras son con referencia al cero de pieza (X_p, Y_p, Z_p), y las últimas correspondientes al cero de pieza (X_{cp}, Y_{cp}, Z_{cp}). Dentro de esta misma sección existen casillas dedicadas al monitoreo de ciertos parámetros como son: el avance, frecuencia y revoluciones por minuto (RPM's) a las que funcionan los actuadores de la MMH.

Por último, la sección 9 aloja tres botones, el primero de ellos ejecuta la rutina de origen (*HOME*). Esta rutina consiste en regresar cada uno de los carros a la posición cero de máquina, por razones de seguridad primero se desplaza el eje Z, enseguida el eje X, y por último el eje Y. Estos movimientos cesarán hasta detectar la señal arrojada por los sensores de contacto eléctrico (*push buttons*). Después de haber sensado este regreso se ejecuta una rutina de 400 pasos en dirección contraria para cada uno de los ejes, esto es como medida de seguridad.

También es importante mencionar que cada vez que exista un cambio de dirección en cualquier eje se debe ejecutar una rutina que compense el juego mecánico (*backlash*). El botón rotulado como *ALTO* envía una cadena que al ser descifrada por el microcontrolador cesa cualquier tipo de movimiento, almacenando el estado actual en el que quedaron energizados los actuadores. El botón *SALIR* tiene la misma función que en la pantalla de inicio.

Una vez que se han descrito las secciones elementales de la interfaz de usuario, se proponen unos ejemplos que muestren el funcionamiento del programa de control.

4.9 Ejemplos del proceso de control.

La forma de procesar una cadena de control mediante este sistema, se ejemplifica con dos ejemplos. Estos dos ejemplos fueron propuestos debido a que son representativos de los cuatro modos de operación del sistema. La *Cadena1* ejemplifica los modos *joystick* y por *pasos* (modos muy similares), mientras que la *Cadena2* ejemplifica los modos por *línea* y por *programa* (modos similares). Las cadenas propuestas son las siguientes:

- a) Cadena1: JX+;
- b) Cadena2: G02X02969.00Y03400.00I03012.00J03040.00F08;

Estas cadenas son enviadas por la interfaz de usuario y recibidas vía puerto serie por la tarjeta de control, a través del módulo *USART (Universal Synchronous Asynchronous Receiver Transmitter)* del microcontrolador PIC.

4.9.1 Ejemplo (a).

Una vez recibida la *Cadena1* (JX+) mediante la atención a una interrupción, se almacena en un vector llamado *CadenaControl* de 35 elementos. El primer carácter de la cadena es almacenado en el elemento cero del vector, el segundo en el elemento uno y así sucesivamente. En el momento en el que se detecte el carácter fin de línea (;) se termina el proceso de almacenamiento. La forma de procesar la *Cadena1* una vez almacenada se muestra en el diagrama de la figura 37.

Para decodificar la cadena de control, se ejecuta la rutina de modalidad, ésta lee el primer elemento del vector y descifra el modo en el cual se quiere operar la MMH. En este caso, la letra *J* indica que los desplazamientos se harán por medio de un *joystick*. Aparte de descifrar la modalidad de desplazamiento, genera la ruta de procesamiento correspondiente a esa modalidad, es decir, predice que parámetros son los próximos a decodificar. Para este caso en particular, se espera que se defina el grado de libertad en el cual se desplazara la MMH, para lo cual se invoca a la rutina *obténde eje*. Cabe aclarar que si el carácter leído no se puede descifrar, es decir, no se encuentra dentro de la rutina esperada, el programa interpreta que la sintaxis de la cadena de control es errónea y no ejecuta tarea alguna.

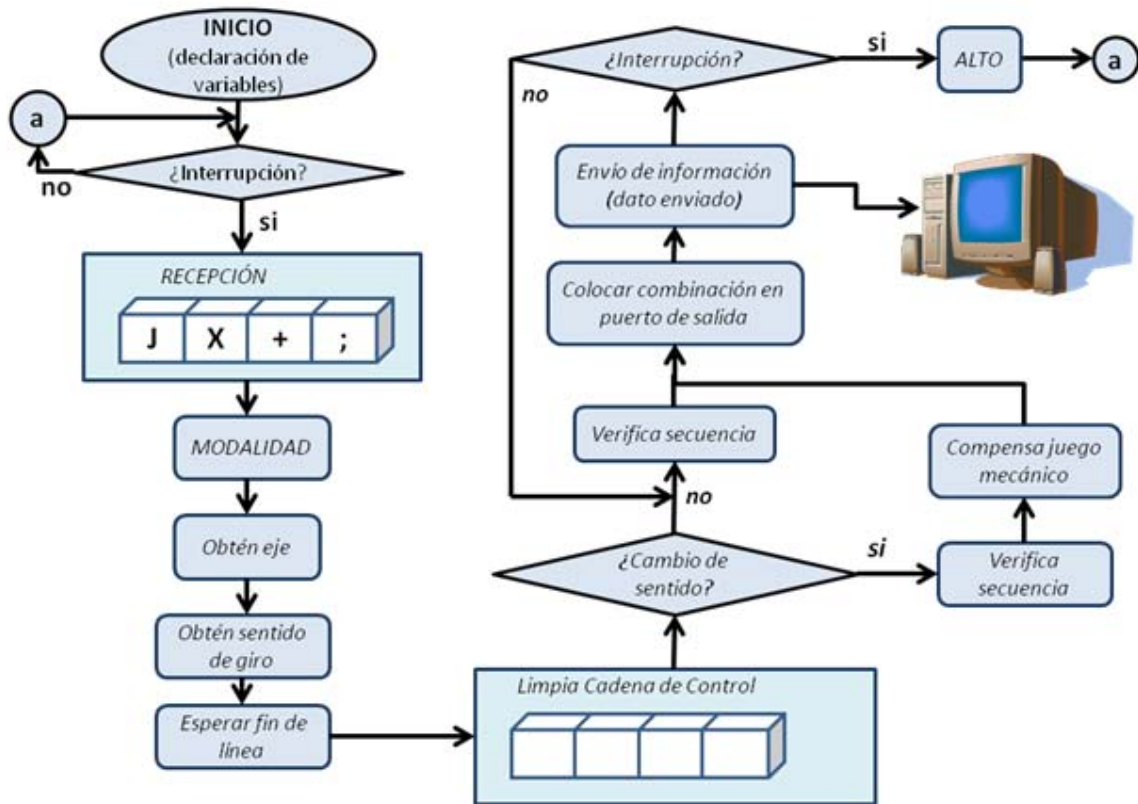


Figura 37. Proceso de control para la cadena "JX+;".

Después de que el programa ha descifrado que los desplazamientos se harán en el eje X, el proceso necesita saber el sentido de giro del actuador para lo cual se ejecuta una rutina llamada *obten sentido de giro*. En esta rutina se pregunta si el sentido corresponde al carácter “-” (guión medio), el cual es interpretado como un sentido antihorario, en caso de no ser así, el programa interpreta que se trata de un sentido horario. Para finalizar esta decodificación se ejecuta la rutina *esperar fin de línea*, la cual espera el carácter “;” (punto y coma) y de no ser así, el código es erróneo.

Ya que ha sido descifrada la modalidad de control se limpia el vector *CadenaControl* para una nueva recepción, y se ejecuta inmediatamente después la rutina que pregunta si hay *cambio de sentido*. En esta rutina se compara el estado de dos banderas: *sentidoactual* y *sentidoanterior*. El propósito de esta comparación es compensar el juego mecánico de la MMH, por lo tanto, si el estado de estas dos banderas resulta ser diferente entonces se ejecuta una rutina que compensa el juego mecánico.

Para cualquier tipo de movimiento a realizar en la MMH es obligatorio ejecutar una rutina denominada *verifica secuencia*. Ésta se encarga de seguir un orden para energizar las fases del actuador con el propósito de no ejecutar pasos de más, o avanzar en sentido incorrecto.

La combinación generada (byte) se graba en el puerto D (salidas a actuadores) del microcontrolador, esta combinación permanece grabada en el puerto durante un determinado tiempo (temporización), este tiempo está de acuerdo a la frecuencia en la que se esté ejecutando la modalidad en cuestión. En el modo *joystick* la frecuencia de activación de los actuadores es de 170Hz.

Una vez que el actuador dió el paso en la dirección y sentido correctos, se ejecuta una rutina denominada *envío de información* en la cual informamos a la interfaz, mediante un variable denominada *dato enviado*, que se ha efectuado un movimiento. En la variable usada para este propósito va implícita la dirección y sentido del movimiento. Cabe mencionar que cualquier avance realizado en la MMH será transmitido a la interfaz de usuario para su correspondiente registro: coordenadas y representación gráfica.

Por último, se verifica el estado de la bandera de interrupción por recepción serial para una posible decodificación de una nueva cadena de control. En caso de estar activada se detiene el avance de cualquier actuador mediante la rutina *Alto*, y se vuelve a ejecutar la rutina de *Recepción*. De no estar activada la bandera de interrupción, se prosigue en la siguiente combinación para energizar el actuador mediante la rutina llamada *verifica la secuencia*.

4.9.2 Ejemplo (b).

Para este ejemplo, la cadena a manejar será un código G correspondiente a una interpolación circular en sentido horario cuya sintaxis es como sigue: *G02X02969.00Y03400.00I03012.00J03040.00F08*; El procedimiento a seguir es mostrado en la figura 38 y es muy parecido al del ejemplo (a), con la diferencia que algunas rutinas cambiarán debido a que esta modalidad requiere de parámetros diferentes.

Hasta el bloque de modalidad todo resulta igual. Después de éste, se genera obviamente una ruta diferente ya que la modalidad descifrada es distinta, la cual en este caso es *línea de código G*. En consecuencia, se invoca una rutina llamada *obtención de 2 dígitos*, esta rutina es la encargada de descifrar y almacenar el número de código G en una variable denominada *NoCodigoG*, para después pasar a la rutina de *obtención de eje* donde el procedimiento a seguir resulta ser el mismo que en el ejemplo anterior, hasta el bloque de *sentido de giro*.

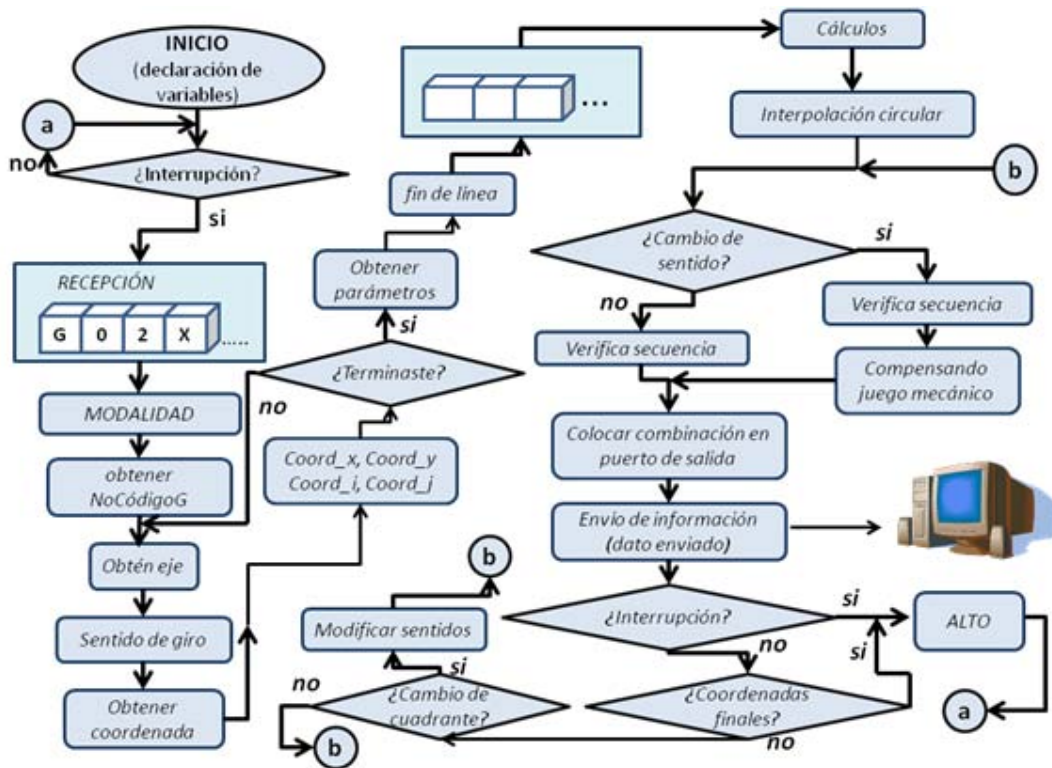


Figura 38. Proceso de control para un código G02.

Una vez que se ha determinado el eje y el sentido de giro, se procede a obtener la magnitud de la coordenada en cuestión, esto se logra a través de dos rutinas, una para obtener la magnitud entera de cinco dígitos y otra para obtener la magnitud decimal, de dos dígitos. Debido a que este código G requiere de 4 coordenadas en total, los procesos de obtención de eje, sentido de giro, obtención de magnitud entera y decimal se repiten cuatro veces.

Después del bucle generado para la obtención de coordenadas, el programa ejecuta la rutina *obtención de parámetros*. En ésta, se tiene la posibilidad de descifrar algunos parámetros como son: avances, revoluciones por minuto, herramienta, y pausas. En el caso del código a tratar (G02) se requiere el parámetro *avance*, para lo cual se ejecuta la rutina de obtención de 2 dígitos, esta rutina almacena la magnitud de este parámetro en una variable denominada *Avance*.

Sin más parámetros requeridos para esta interpolación se verifica el carácter fin de línea, y en caso de darse se limpia el vector *CadenaControl* y se pasa a una rutina de cálculos, propios de la interpolación a realizar.

La rutina de cálculos dicta las condiciones iniciales bajo las cuales comenzará el método de interpolación (cuadrante inicial, cuadrante final, radio, frecuencia, distancias, etc.). Para el caso de interpolación circular el método a emplear es el de *Búsqueda Directa*. Durante la ejecución de éste método de interpolación se estarán procesando continuamente los bloques restantes, a partir del bloque *cambio de sentido* hasta el bloque para verificar el estado de la bandera por interrupción.

Por último, después de ejecutarse cada iteración del algoritmo, se verifica si alcanzo las coordenadas finales y si hay cambio en el número de cuadrante, con el propósito de parar la ejecución del algoritmo de interpolación ó continuar con el mismo, modificando sentidos de giro.

4.10 Hardware electrónico.

Durante el desarrollo de sistemas de control, el GMM ha probado con diferentes circuitos integrados (C.I.) de potencia con el fin de tener el mejor accionamiento en los actuadores de las MMH's. En este trabajo, se proponen dos diferentes C.I. de potencia, el PBL3717A de la compañía *ST Microelectronics* y el A3988 de *Allegro MicroSystems*. La selección del C.I. a emplear es realizada con el mismo método empleado para el algoritmo de interpolación (matriz de decisión). Para la aplicación de éste método se practican los pasos enunciados en el apartado *Comparación de los métodos de interpolación y Selección del método de interpolación* de este capítulo.

4.10.1 Circuito PBL3717A.

El PBL3717A es un circuito capaz de controlar una fase de un motor a pasos bipolar, este circuito cuenta con control de corriente (por chopper) que limita ésta última a tres niveles seleccionables a través de dos entradas lógicas, I_0 e I_1 (pines 7 y 9). Cuando estas dos entradas están a un nivel alto el dispositivo está deshabilitado.

La sección de potencia está compuesta por un puente H bipolar con cuatro diodos internos de conmutación rápida, los cuales protegen los devanados del motor de posibles regresos de corriente. Este circuito es capaz de proporcionar una corriente de hasta 1.2A, con un voltaje de carga que puede ir de los 10 a los 50V. El diagrama interno que describe el funcionamiento de este circuito es mostrado en la figura 39.

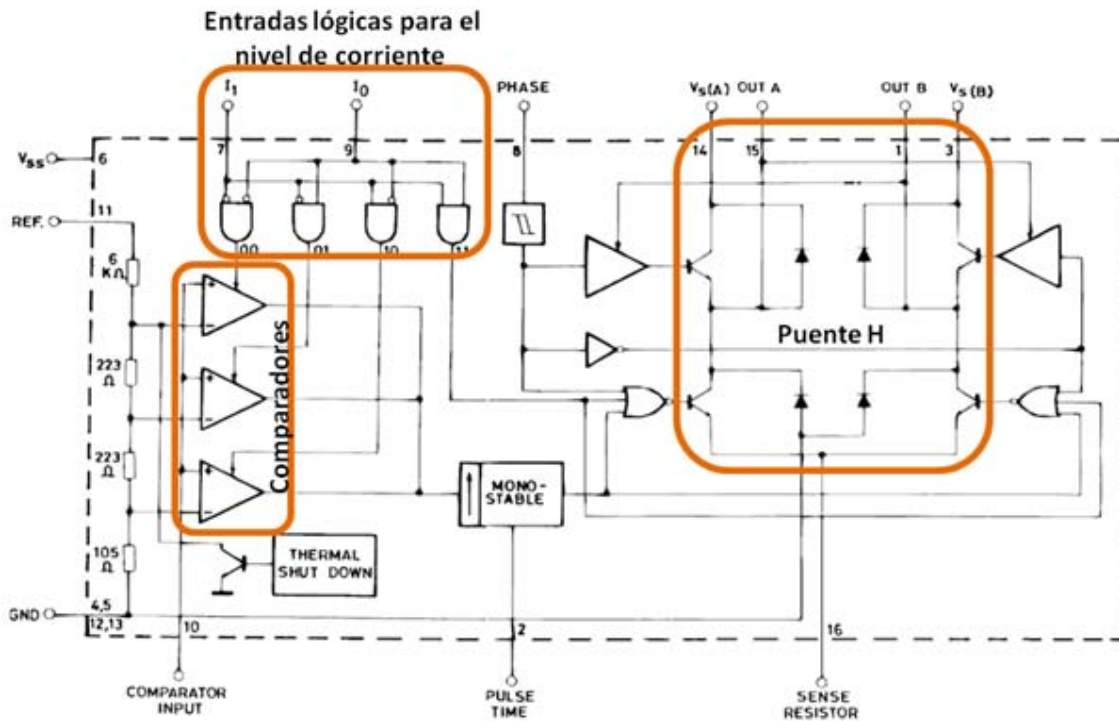


Figura 39. Circuito interno del PBL3717A.

A este tipo de circuito se le denomina de *tiempo de desconexión constante*, es decir, una vez alcanzado el valor de la corriente del devanado (previamente establecida), ésta se interrumpe durante un tiempo fijo. Una red externa RC conectada al pin 10 determina el tiempo de apagado de los transistores de potencia inferiores del puente H, tal y como lo describe la ecuación 4.1.

$$T_{OFF} = 0.69RC \dots\dots\dots (4.1)$$

Esta magnitud representa el tiempo que se interrumpirá la corriente por el devanado una vez que haya alcanzado su valor prefijado. Así pues, Los pulsos detectados por los transistores de potencia son generados por un circuito monoestable que se encuentra a la salida de los comparadores.

Los valores de las entradas lógicas I_0 e I_1 , y el valor del voltaje de referencia del pin 11, determinan el voltaje a compararse con el voltaje generado desde la resistencia de sensado en el pin 16, que a su vez está conectada a los emisores de los transistores inferiores de cada puente H.

Cuando la corriente por el devanado sobrepasa el valor a controlar, un comparador dispara al monoestable que desconecta los transistores de potencia inferiores durante el tiempo determinado por la ecuación 4.1. El valor de la corriente pico máxima a controlar se determina por la siguiente expresión:

$$I_{pico} = \frac{0.083V_{ref}}{R_{sen}} \dots\dots\dots (4.2)$$

donde:

I_{PICO} : Corriente pico máxima.

V_{REF} : Voltaje de referencia.

R_{SEN} : Resistencia de sensado.

El sentido en el que circula la corriente en el devanado, se controla por medio de la entrada digital *FASE* (pin 8). El circuito mostrado en la figura 40, permite operar un motor a pasos bipolar de manera bidireccional y de acuerdo a la secuencia formada por las señales de entrada *FASE A* y *FASE B* (*PhA* y *PhB*).

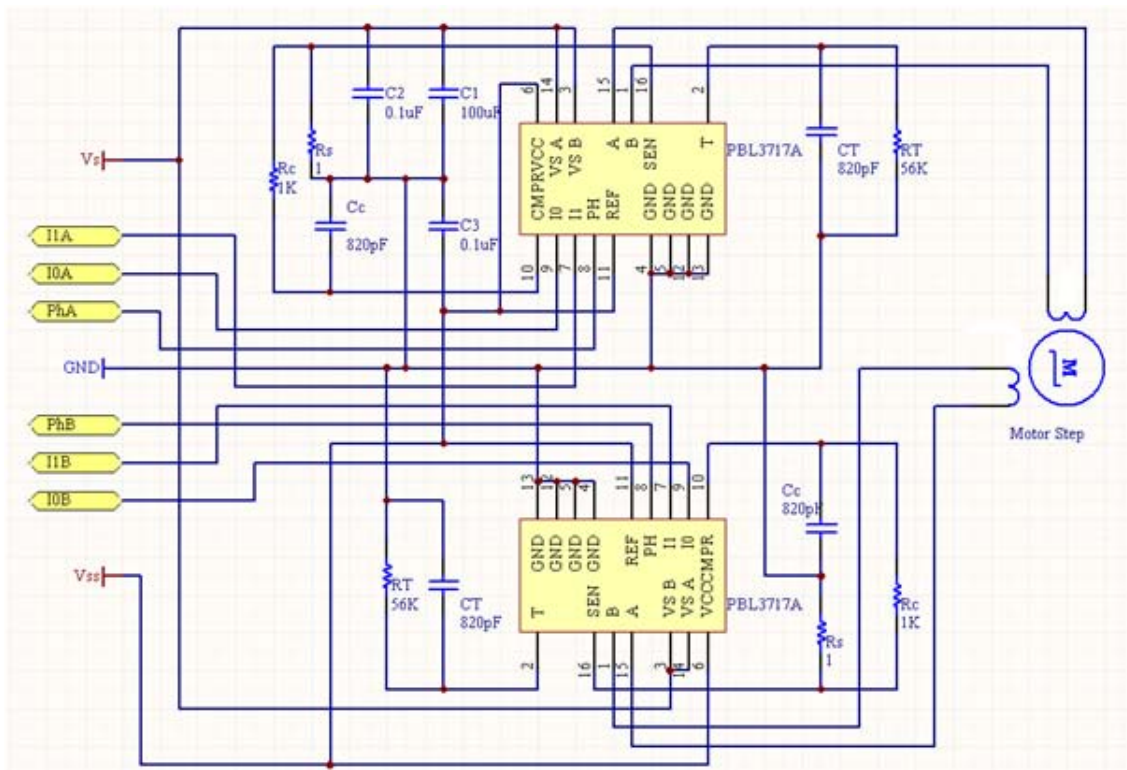


Figura 40. Circuito para controlar un motor a pasos bipolar de manera bidireccional.

El PBL3717A permite implementar micropasos, esto significa emplear varios niveles de corriente para obtener ángulos de desplazamiento pequeños (control de posición más preciso), pero a expensas de un menor torque. De tal forma, el motor a pasos puede operar en modo de *paso completo*, *medios pasos* y *cuartos de paso*. En el modo de *paso completo*, ambas fases se energizan con la misma magnitud de corriente, y mientras I_0 e I_1 permanezcan fijas el torque será constante en cada paso. En *medios pasos*, la alimentación de las fases es de forma alternada por lo que hay determinadas combinaciones en donde una fase no se energiza, debido a esto el torque es menor que en el modo de *paso completo*.

Para la operación en *cuartos de paso* el motor tiene que funcionar a máxima corriente para evitar pasos irregulares. La secuencia correcta para energizar las fases en cualquiera de los tres modos descritos anteriormente es mostrada en la tabla 10. Para controlar este circuito, las entradas lógicas I_{0A} , I_{1A} , Ph_A , I_{0B} , I_{1B} y Ph_B mostradas en la figura 40, son generadas a través de la tarjeta controladora (PIC18F8720).

Modo de operación	Secuencia
Paso completo	$A\bar{B} \rightarrow AB \rightarrow \bar{A}B \rightarrow \bar{A}\bar{B}$
Medios pasos	$A\bar{B} \rightarrow A \rightarrow AB \rightarrow B \rightarrow \bar{A}B \rightarrow \bar{A} \rightarrow \bar{A}\bar{B} \rightarrow \bar{B}$
Cuartos de paso	$AB \rightarrow \frac{A}{2}B \rightarrow B \rightarrow \frac{\bar{A}}{2}B \rightarrow \bar{A}B \rightarrow A\frac{B}{2} \rightarrow \bar{A} etc.$

Tabla 10. Secuencias para los diferentes modos de operación del PBL3717A.

Por último, en la entrada FASE de cada circuito se incorpora un *schmitt trigger* con el fin de proporcionar inmunidad al ruido. Además, este circuito cuenta con una protección por exceso de temperatura que inhibe la operación a máxima corriente. El encapsulado de este circuito es un DIP16 de dimensiones 20x9mm a un precio de aproximadamente \$2USD.

4.10.2 Circuito A3988.

El A3988 es un circuito diseñado para operar dos motores a pasos bipolar ó cuatro motores de DC. La corriente máxima es de 1.2A con un voltaje de carga que va de los 8 a los 36V. Esta corriente es proporcionada por cada uno de los puentes H (de transistores DMOS) y regulada a través de un circuito interno PWM, éste usa un circuito disparador que controla el tiempo que los *drivers* permanecen apagados, internamente este tiempo es fijado a 30µs. La corriente pico en cada puente es determinada a través de resistencias sensoras (R_{SENS1} , R_{SENS2} , R_{SENS3} y R_{SENS4}) y de un voltaje de referencia (V_{REF}). El diagrama interno de éste circuito y su conexión a motores de pasos es mostrado en la figura 41.

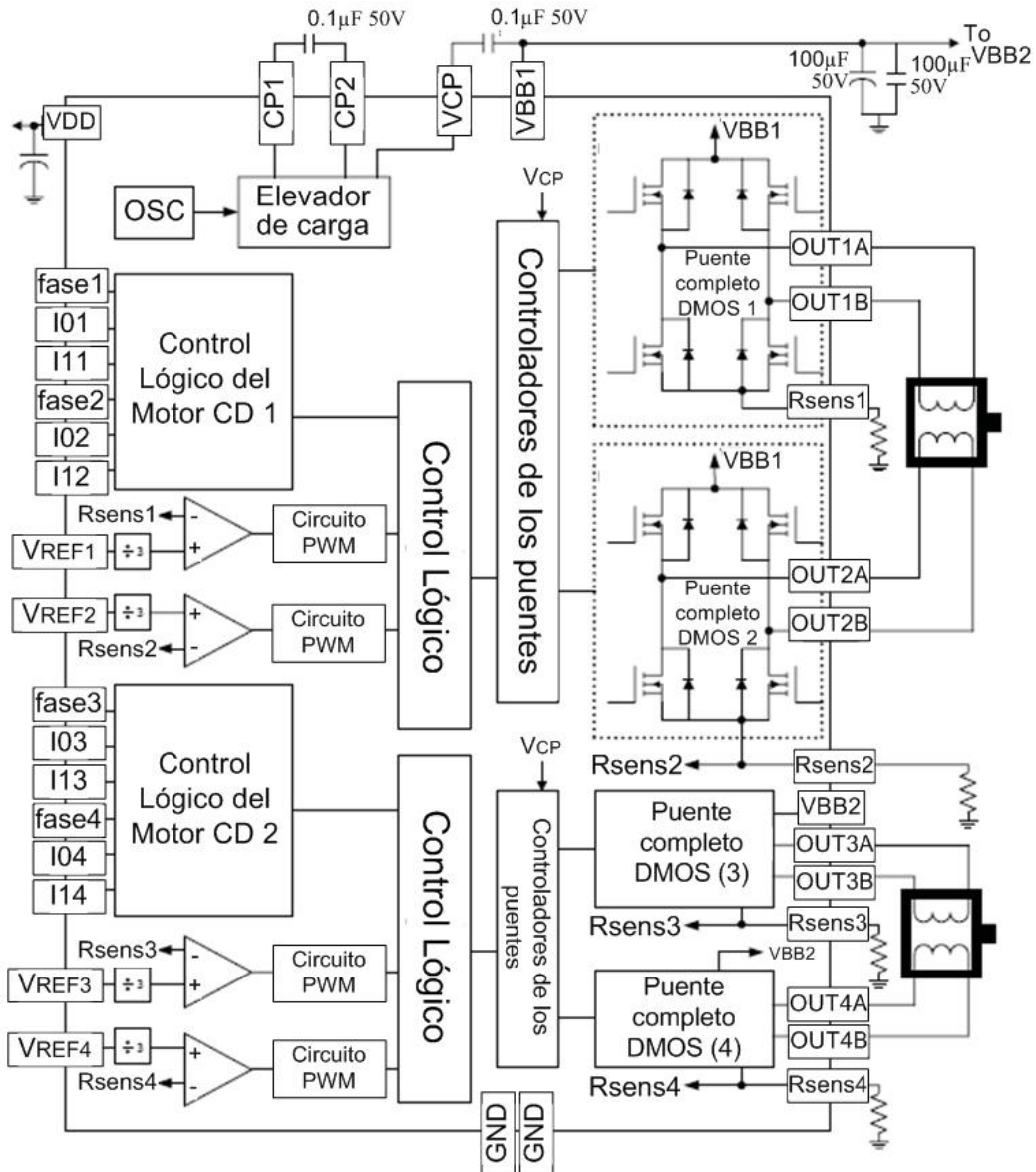


Figura 41. Circuito interno del A3988 para controlar dos motores a pasos bipolares.

Las entradas lógicas de este circuito son las mismas que en el caso del PBL3717A (*PHASE*, I_o e I_i) y se recomienda que de ser alimentadas a un nivel lógico alto se haga a través de resistencias de *pull-up*, con el fin de limitar la corriente de entrada y evitar sobrevoltajes. Además, la configuración de estas entradas permite establecer los modos de funcionamiento del motor: paso completo, medios pasos y cuartos de paso.

Al igual que el circuito anterior, el A3988 cuenta con un circuito que al detectar un sobrecalentamiento deshabilita las salidas del dispositivo hasta que las condiciones mejoren. Este circuito se tiene en dos matrículas diferentes, A3988EV y A3988JP, ambas son de montaje superficial con un área de contacto expuesta para un mejor desempeño térmico. El A3988EV es un C.I de 36 pines de 6x6mm, mientras que el A3988JP es un C.I de 48 pines de 7x7mm. El precio aproximado de este circuito es de unos \$5USD.

Una vez descrito el funcionamiento de ambos C.I's, en el siguiente apartado se emplea el mismo método realizado en los apartados 4.5 y 4.6 de este capítulo, para evaluar y seleccionar el circuito de potencia a implementar en el desarrollo de este sistema de control.

4.10.3 Evaluación y selección del circuito de potencia.

En la tabla 11 se ordenan jerárquicamente los parámetros de importancia para la evaluación de los circuitos de potencia. Los valores relativos asignados a los parámetros manejan los mismos niveles de importancia que en la sección 4.5.

Parámetros de importancia	Valor relativo asignado
Corriente proporcionada.	3
Voltaje de carga.	3
Recursos para implementar.	3
Elementos externos requeridos.	1

Tabla 11. Parámetros de importancia en la comparación de circuitos de potencia.

En la tabla 12 se muestra el resultado de la evaluación de los circuitos con respecto a cada uno de los parámetros listados en la tabla 11. En *cursivas* se muestra el resultado de multiplicar la calificación por el factor de importancia de cada parámetro. La forma de realizar esta evaluación fue de la siguiente manera: para la *corriente*, se calificó con 10 a ambos circuitos debido a que la magnitud proporcionada de 1.2A fue suficiente, tomando en cuenta que la resistencia en cada bobina es de 29Ω [28] y el intervalo de voltaje en la operación de los motores a pasos fue de los 8 los 14V. En cuanto al *voltaje*, se le otorgó una mayor calificación al PBL3717A debido a la posibilidad de operarlo en un mayor intervalo de voltaje.

Circuito de potencia	Corriente (3)		Voltaje (3)		Recursos para implementar (3)		Elementos externos requeridos (1)		Total
PBL3717A	10	<i>30</i>	10	<i>30</i>	10	<i>30</i>	8	7	97
A3988	10	<i>30</i>	9	<i>27</i>	5	<i>15</i>	10	<i>10</i>	82

Tabla 12. Evaluación de los circuitos de potencia.

El parámetro denominado *recursos para implementar* se refiere a los recursos (infraestructura) con los que cuenta el GMM para llevar a cabo la prueba del circuito en cuestión, esto sin duda alguna favoreció al PBL3717A debido el tipo de encapsulado que maneja. Mientras que el circuito A3988 debido a su tamaño y encapsulado, dificultó la implementación para desarrollar pruebas.

Por último y referente a los *elementos externos requeridos* por cada circuito, el A3988 tal como se muestra en la figura 41, requiere de un solo encapsulado para controlar dos motores a pasos y 5 capacitores mas 4 resistencias. En cuanto al PBL3717A, se requieren de 4 C.I. para controlar la misma cantidad de motores a pasos y aproximadamente cinco veces más en componentes como capacitores y resistencias.

Como resultado de la evaluación realizada en la tabla 12, el circuito PBL3717A es implementado como el sistema de potencia del sistema de control propuesto. En la figura 42 se muestra una imagen del circuito impreso correspondiente a una etapa de potencia para dos motores a pasos. Esta tarjeta consta de 4 conectores molex (de cuatro y tres pines), uno para la alimentación, otro para las señales de control provenientes de la tarjeta controladora (Microcontrolador PIC), y dos más para las salidas a los motores a pasos. También cuenta con un circuito de aislamiento (PC847) con el fin de proteger a la tarjeta controladora. De tal forma, esta tarjeta pretende mejorar los problemas relacionados con la limitante respecto al número de actuadores controlados por el sistema de control [9].



Figura 42. Sistema de potencia para dos actuadores de la MMH.

Es importante mencionar que el C.I A3988 es una muy buena opción a futuro, es decir, la posibilidad de utilizar C.I de montaje superficial reducirá enormemente el espacio ocupado por el sistema de control. Por el momento este C.I resulta poco conveniente debido a problemas relacionados con el proceso de fabricación del circuito impreso, la sujeción del componente y la aplicación de la soldadura en las respectivas terminales.

El esquemático y diseño del circuito impreso de la tarjeta mostrada en la figura 42, están reportados en la sección *A7 de los Anexos* de este trabajo. En el siguiente capítulo se describen las pruebas realizadas a todo el sistema de control propuesto en este trabajo, haciendo énfasis en los consumos, magnitudes, errores y tiempos obtenidos.



Pruebas y resultados.

En este capítulo, se presentan las pruebas que se realizaron al sistema de control descrito en los cuatro capítulos anteriores. Primeramente se obtuvieron las señales generadas por la tarjeta controladora, y el circuito de potencia seleccionado. Después se realizaron tres tipos de pruebas al sistema micromecánico en cuestión. La primera prueba consistió en maquinar figuras simples como rectas y cuadrados, en donde los movimientos realizados son en un eje a la vez. El objetivo de esta prueba es determinar el juego mecánico, que existe en los ejes X y Y de la MMH, y compensarlo de la mejor manera posible.

En la segunda prueba, se maquinaron figuras que requerían interpolación en dos ejes (circunferencias concéntricas), mientras que el tercer tipo de pruebas, consistieron en la ejecución de programas en código estándar, G y M.

Para el desarrollo de las pruebas enunciadas anteriormente, se emplearon los siguientes elementos: una PC con Windows 2000 y puerto serie, la MMH de la figura 24, láminas delgadas de acero y latón bañadas en tinta azul para metal, como herramienta de desbaste fue empleada una punta de acero, un osciloscopio de cuatro canales, una fuente de alimentación dual de 30V, la tarjeta controladora mostrada en la tabla 5 y el circuito de potencia de la sección de *A7 de los Anexos* implementado en tarjeta de prototipos. El sistema propuesto y cada una de sus partes se muestran en la figura 43. El maquinado de las figuras comienza a partir del momento en el que la punta hace contacto con la lámina, para después comenzar con el seguimiento de las trayectorias establecidas dentro del código programado.



Figura 43. Sistema propuesto para las pruebas al sistema de control.

Las mediciones de los maquinados realizados en este trabajo se llevaron a cabo por medio de un comparador óptico de marca “*Nikon Profile*” mostrado en la figura 44, que se encuentra en el Laboratorio de Metrología del CCADET, UNAM.

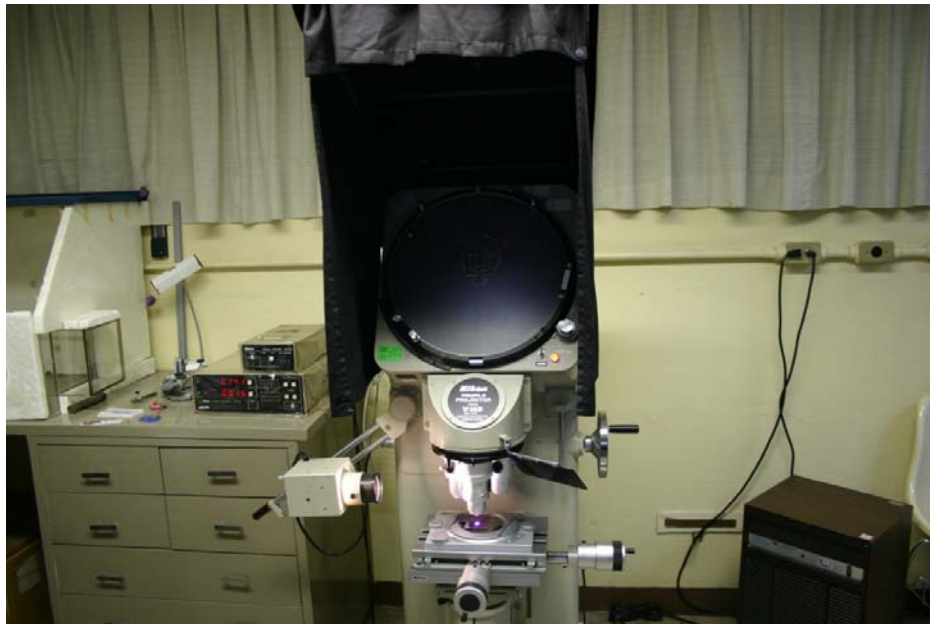


Figura 44. Comparador óptico del Laboratorio de Metrología.

5.1 Pruebas realizadas al *hardware* electrónico.

Las pruebas realizadas al *hardware* electrónico consistieron primeramente en la obtención de las señales generadas por la tarjeta de control. En la figura 45, se muestran los trenes de pulsos generados para desplazar los carros en cada uno de los ejes X, Y y Z de la MMH, a una frecuencia de 42Hz con un de voltaje lógico TTL.

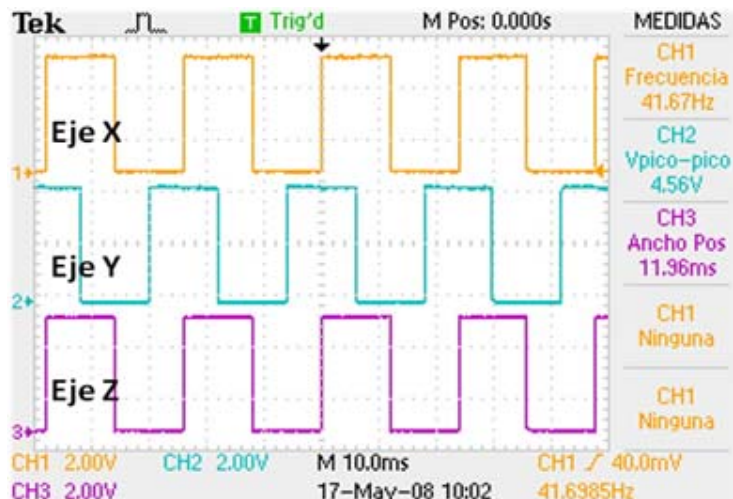


Figura 45. Señales generadas por la tarjeta de control para los tres ejes de la MMH.

Por otro lado, en la figura 46 se observan las señales generadas por el sistema de potencia (sin carga) al proporcionarle las señales de la figura 45, con un voltaje de carga de 11.5V registrándose un consumo de todo el sistema de 1.2A (tres grados de libertad).

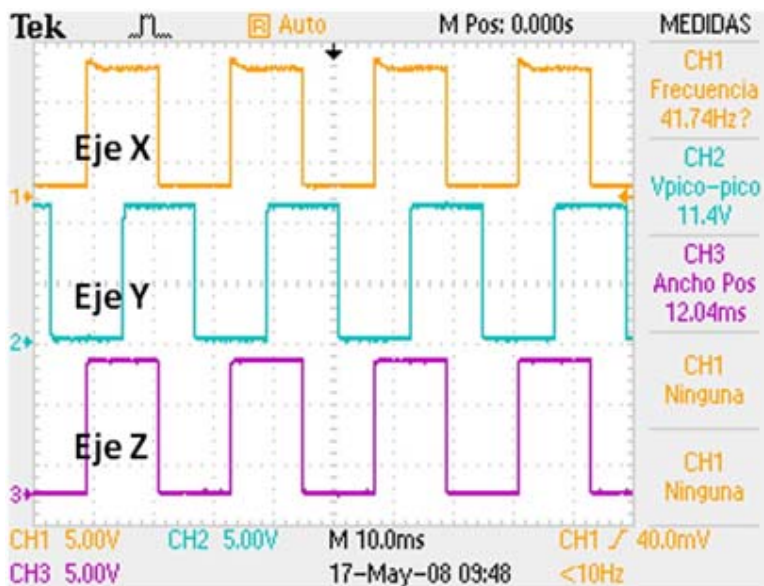


Figura 46. Señales generadas por la etapa de potencia (sin carga).

En la figura 47, se tienen las señales generadas por el sistema de potencia al conectarles los motores de la MMH, El consumo de corriente al energizar el sistema a 8V fue de 0.9A, mientras que el consumo registrado a los 10V fue de 1.1A.

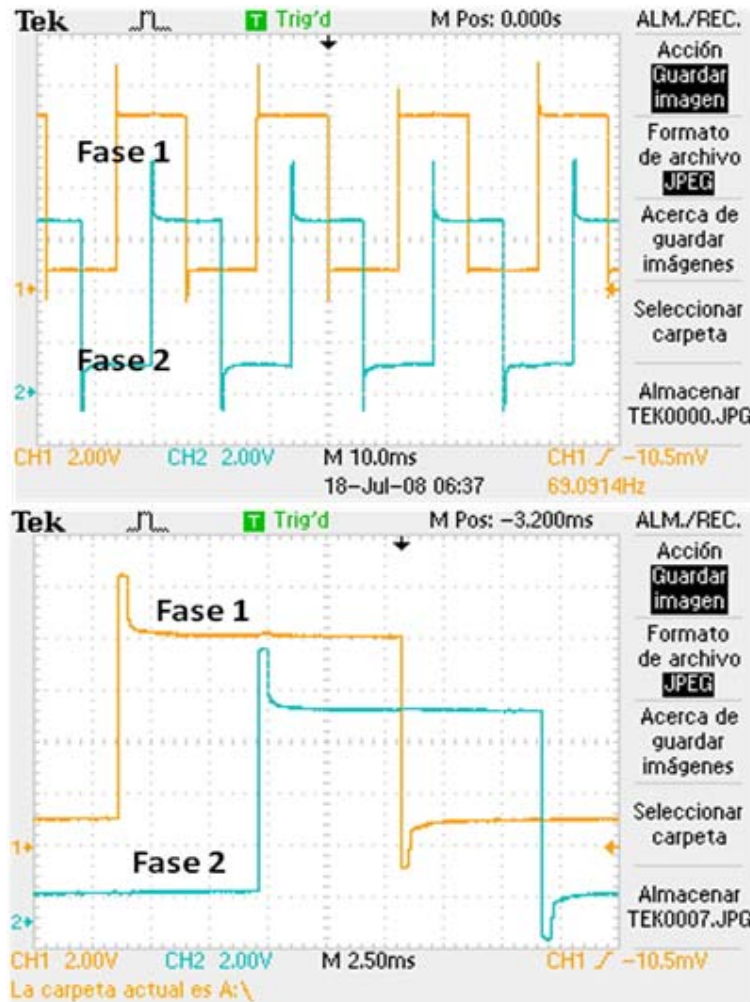


Figura 47. Señales generadas por la etapa de potencia con carga inductiva.

En estas señales se aprecian unos picos debido a efectos inductivos provocado por las fases del motor al ser conectado al sistema de potencia. Para mejorar estas señales se propone un circuito LC, mostrado en la figura 48, a la salida de cada una de las fases de un motor (circuito PBL3717A). Para determinar el valor de la inductancia y capacitancia del arreglo LC, se aplican las siguientes ecuaciones:

$$L \approx \frac{1}{10} L_M \dots\dots\dots (5.1)$$

$$C \approx \frac{4 \cdot 10^{-10}}{L} \dots\dots\dots (5.2)$$

donde:

L_M : Inductancia de la fase del motor.

Aplicando las ecuaciones (5.1) y (5.2), para un valor de $L_M = 18\text{mH}$ [28], se obtuvo una inductancia $L = 1.8\text{mH}$, y consecuentemente una $C \approx 220\text{nF}$. Las señales obtenidas en las fases del motor son mostradas en la figura 49.

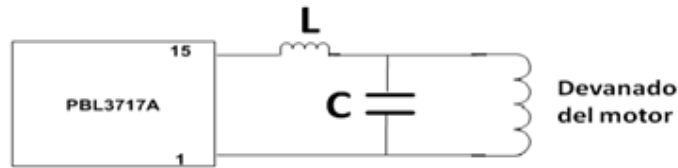


Figura 48. Circuito LC para reducir efectos inductivos en las fases del motor.

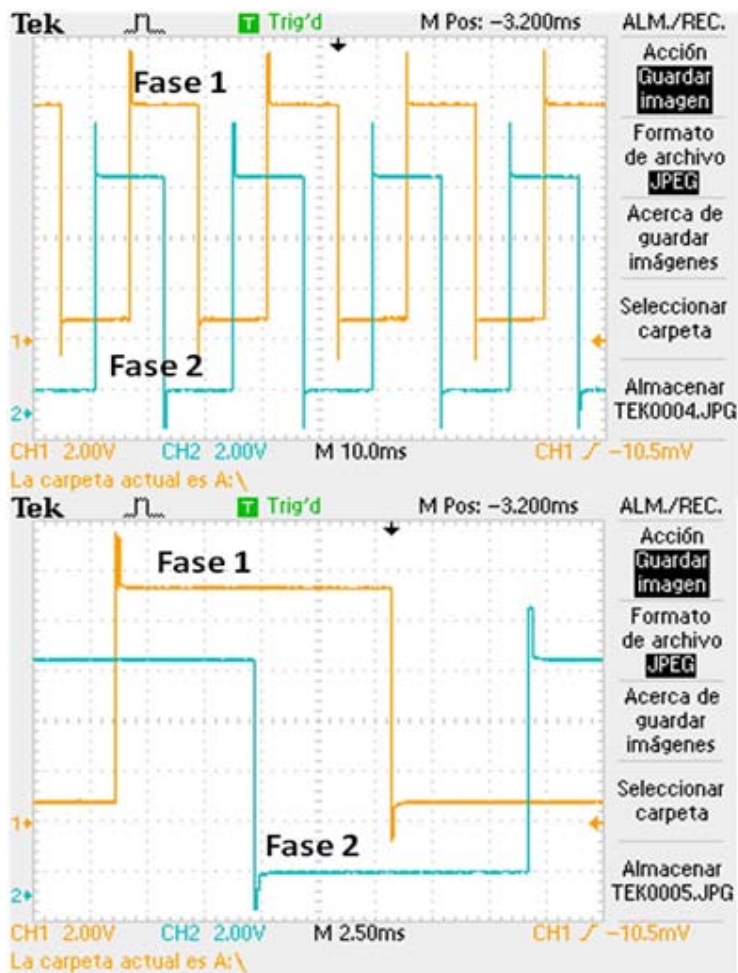


Figura 49. Señales generadas por el sistema de potencia con circuito LC.

Después de haber implementado el arreglo LC, se puede apreciar en la imagen (de mayor periodo) de la figura 49, que el área formada por los picos disminuyó, es decir los picos son más angostos. Como parte de este ejercicio, se implementó un arreglo LC doble en serie (segundo orden) en una sola fase del motor (fase 1), mientras que en la fase 2 no se tiene circuito LC. Las señales obtenidas son mostradas en la figura 50.

En estas imágenes se observa como se reduce, aún mas, el área formada por los picos, al aumentar el orden del arreglo. Esto representa un menor consumo de corriente y menos calentamiento en los devanados del motor.

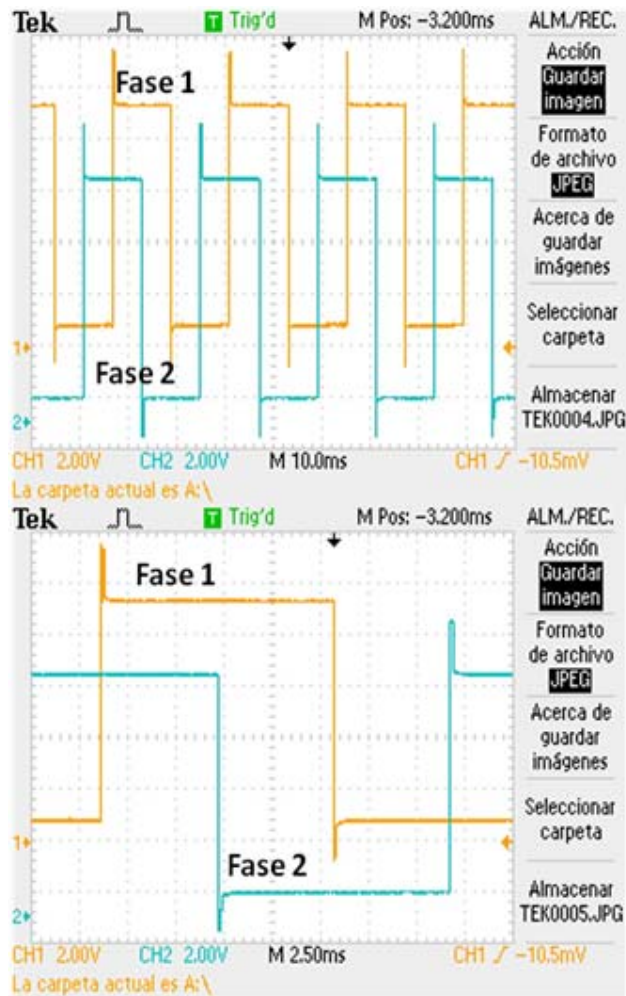


Figura 50. Señales generadas con circuito LC de segundo orden en fase 1.

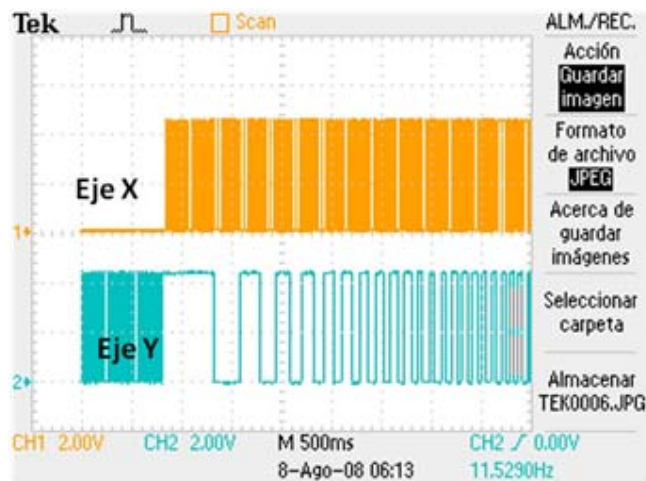


Figura 51. Señales generadas por un código G02.

Por último, en la figura 51 se tienen registradas las señales de los ejes X y Y, al procesar un código G02 (interpolación circular en sentido horario). Estas señales corresponden a la transición del cuadrante 2 al cuadrante 3 en una circunferencia de un radio de 100µm.

5.2 Maquinado de figuras simples.

Los primeros maquinados llevados a cabo consistieron en dos rectas paralelas a cada uno de los ejes en cuestión X y Y, unidas en uno de sus extremos por otra recta. Con el fin de medir el juego mecánico en toda el área de trabajo, se dividió está última, en tres regiones horizontales (H1, H2 y H3) y tres verticales (V1, V2 y V3). Las dimensiones del área de trabajo son: 10mm en el eje X y 7mm en el eje Y.

Así pues, la diferencia de las longitudes de las dos rectas, nos permitirá tener una mejor evaluación del juego mecánico existente en cada una de las tres regiones verticales y horizontales del área de trabajo. En la figura 52, se muestran los resultados obtenidos de estos maquinados mientras que, en la tabla 13 se muestran los datos arrojados por estas pruebas.

Tal como lo muestra la figura 52, así como las magnitudes A, B, C, D, E y F colocados en la tabla 13, el juego mecánico no es uniforme en toda el área de trabajo. De tal forma, se obtuvo la media aritmética (para X y Y) con estos valores a través de la ecuación (5.3), resultando un juego mecánico por compensar para el eje X de 192.5µm, mientras que en el eje Y fue de 168.5µm.

$$\bar{X} = \frac{1}{N} \sum_{i=1}^N x_i \dots \dots \dots (5.3)$$

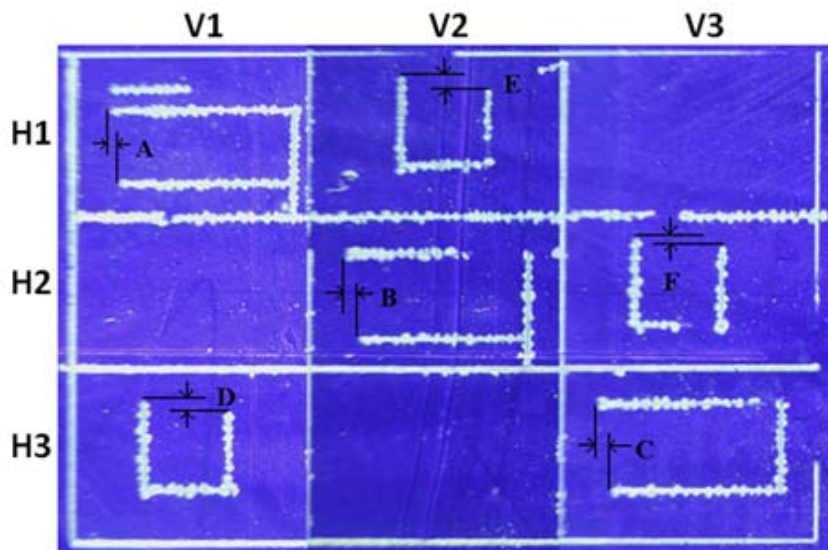


Figura 52. Maquinados para determinar el juego mecánico en toda el área de trabajo.

Figura	Dimensiones teóricas [mm]	Dimensiones experimentales [mm]	Juego mecánico [μm]	Media Aritmética [μm]
A \rightarrow eje X (región H1)	2.5	2.509	190	---
B \rightarrow eje X (región H2)	2.5	2.503	201	---
C \rightarrow eje X (región H3)	2.5	2.502	187	$\bar{X} = 192.5$
D \rightarrow eje Y (región V1)	1.2	1.205	152	---
E \rightarrow eje Y (región V2)	1.2	1.203	180	---
F \rightarrow eje Y (región V3)	1.2	1.199	173	$\bar{Y} = 168.5$

Tabla 13. Magnitudes del juego mecánico para los ejes X y Y.

Con el fin de corroborar los valores de compensación del juego mecánico obtenidos en la tabla 13, se maquinaron tres cuadrados de 2.5mm en las tres diferentes regiones del área de trabajo. En la figura 53 se muestran los cuadrados maquinados, mientras que en la tabla 14 se muestran los resultados de forma cuantitativa.

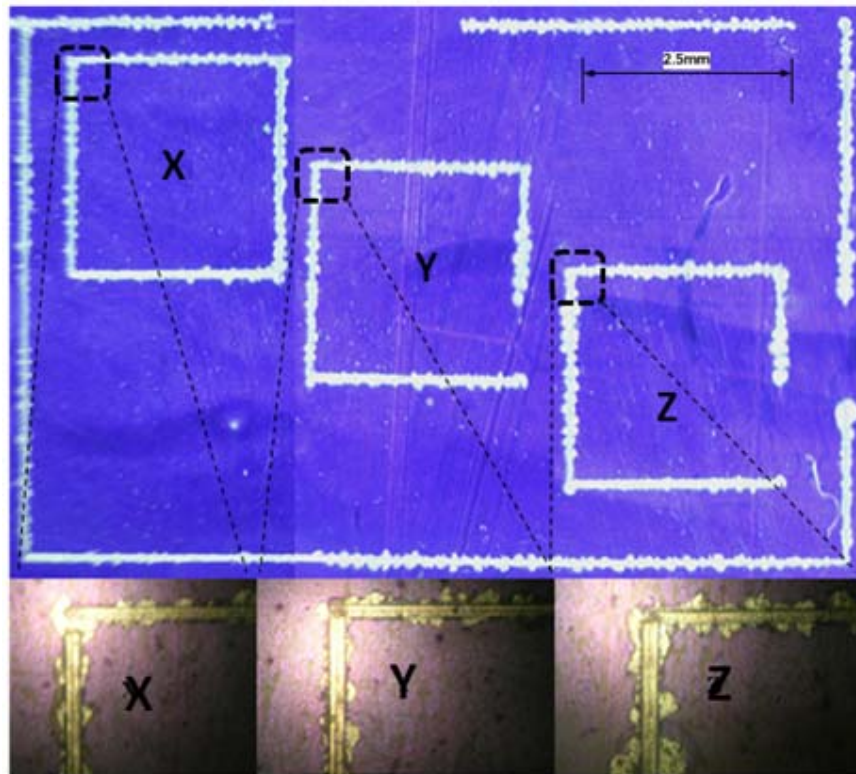


Figura 53. Maquinados para corroborar la compensación del juego mecánico.

Figura	Dimensiones teóricas [mm]	Dimensiones experimentales [mm]	Juego mecánico [μm]	Media aritmética [μm]
Cuadrado_X	2.5	Dx = 2.489 Dy = 2.519	X = 12 Y = 19	$\bar{X} = 11.5$
Cuadrado_Y	2.5	Dx = 2.484 Dy = 2.497	X = 17 Y = 5	$\bar{Y} = 9.5$
Cuadrado_Z	2.5	Dx = 2.497 Dy = 2.494	X = 6 Y = 5	

Tabla 14. Resultados de los maquinados mostrados en la figura 53.

Como se observa en la figura 53, la compensación realizada con el promedio obtenido de los valores de la tabla 13, aún no compensa la totalidad del juego mecánico. De tal forma, se obtuvo de nuevo la media aritmética (ecuación 5.3) de las micras restantes obtenidas en la tabla 14.

En consecuencia, la media aritmética resultante de la tabla 14, en X, es sumada a la ya existente resultando en una nueva compensación en el eje X de $204\mu\text{m}$, mientras que en el eje Y fue de $178\mu\text{m}$. Con estos nuevos valores, se procedió a realizar el maquinado de geometrías que requiriesen la interpolación en ambos ejes.

5.3 Maquinado con interpolación en dos ejes.

El fin de estas pruebas es cuantificar el nivel de aproximación de los algoritmos de control programados en el microcontrolador PIC a una geometría dada, de tal forma que, tanto de manera visual y por dimensionamiento de la geometría propuesta, se comparen los resultados obtenidos en este trabajo con los del sistema de control anterior (*Desarrollo de una interfaz computacional para el control de Micromáquinas-Herramienta*). Esto permite saber si es posible la implementación de estos algoritmos de control en otros prototipos micromecánicos desarrollados.

Para el desarrollo de estas pruebas se propusieron como geometrías de maquinado tres círculos concéntricos (a), (b) y (c) de 3, 2 y 1mm de radio respectivamente. Estas geometrías tienen como fin verificar la capacidad del sistema propuesto para conservar un determinado diámetro, así como el centro de las geometrías. Los maquinados resultantes (realizados en sentido horario) se muestran en la parte alta de la figura 54, y las magnitudes de los diámetros ($\emptyset 1$, $\emptyset 2$, $\emptyset 3$, $\emptyset 4$, $\emptyset 5$, $\emptyset 6$, $\emptyset 7$ y $\emptyset 8$) correspondientes a cada circunferencia son registradas en la tabla 15. Por otro lado, en las figuras 55 y 56 se muestran unos acercamientos correspondientes a las zonas Ps, Pi, Ei y Ed de cada circunferencia de la figura 54. En la figura 55, se observa que la compensación al juego mecánico, del extremo Ei, está sobrada por $4\mu\text{m}$ en las circunferencias (a) y (b) y $2\mu\text{m}$ en la (c). Mientras que en la zona Ed, se compensaron 5, 4 y $2\mu\text{m}$ de más en las circunferencias (a), (b) y (c) respectivamente.

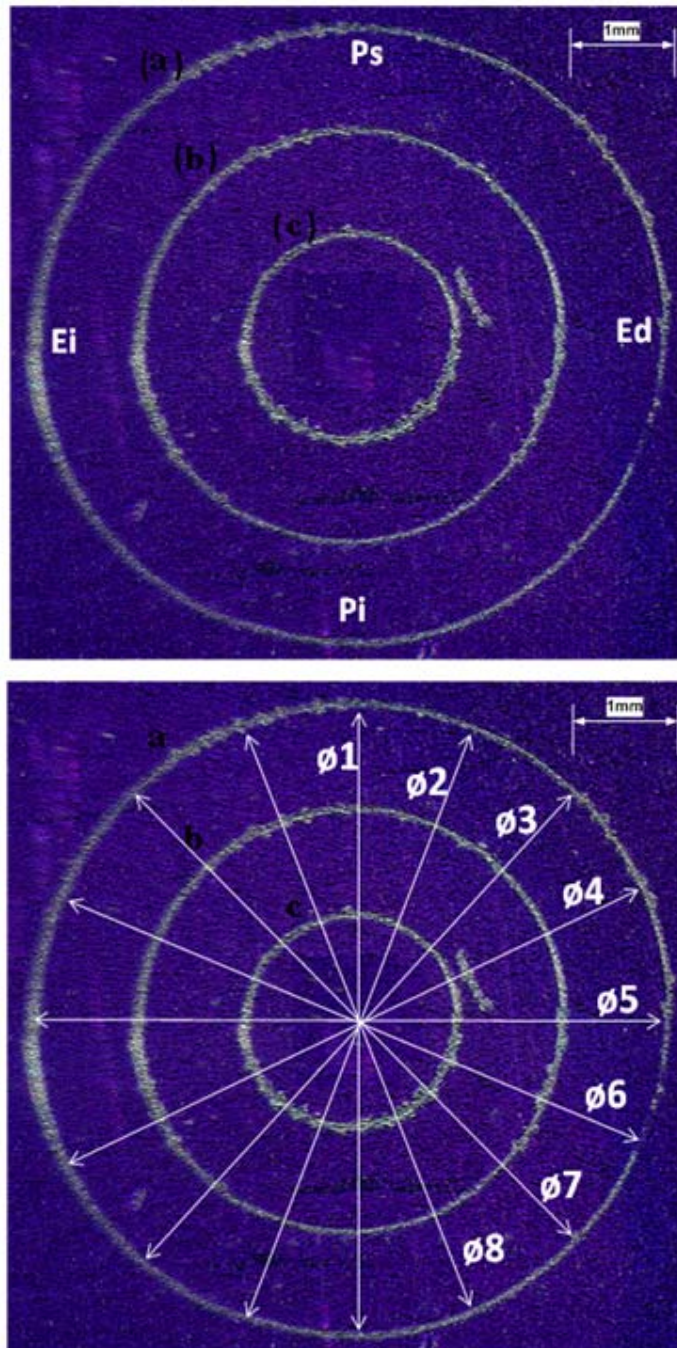


Figura 54. Maquinados con interpolación circular en ejes X y Y.

Por último, en la figura 56 se aprecia que en la zona Pi existe una compensación de más en le eje Y, cuya magnitud es de $2\mu\text{m}$ en la circunferencia (a), mientras que en la zona Ps compensa $1\mu\text{m}$ de mas en las circunferencias (a) y (c).

Circulo	Diámetro teórico [mm]	Diámetros medidos [mm]	Juego mecánico en los extremos [μm]	Tiempo [min]	Medía aritmética y Desviación estándar \bar{X} , σ [μm]
a	$\phi_T = 6$	$\phi_1=5.950$ $\phi_2=5.980$ $\phi_3=6.001$ $\phi_4=6.003$ $\phi_5=5.961$ $\phi_6=5.926$ $\phi_7=5.904$ $\phi_8=5.940$	Ei = 21 Ed = 51 Ps = 8 Pi = 30	23	$\bar{a} = 5.958$ $\sigma_a=0.035$
b	$\phi_T = 4$	$\phi_1=3.987$ $\phi_2=3.991$ $\phi_3=3.998$ $\phi_4=4.018$ $\phi_5=3.990$ $\phi_6=3.961$ $\phi_7=3.947$ $\phi_8=3.969$	Ei = 21 Ed = 25 Ps = 9 Pi = 13	14	$\bar{b} = 3.982$ $\sigma_b=0.022$
c	$\phi_T = 2$	$\phi_1=1.991$ $\phi_2=2.001$ $\phi_3=2.017$ $\phi_4=2.017$ $\phi_5=2.001$ $\phi_6=1.971$ $\phi_7=1.970$ $\phi_8=1.967$	Ei = 18 Ed = 12 Ps = 9 Pi = 3	8	$\bar{c} = 1.991$ $\sigma_c=0.020$

Tabla 15. Resultados de los maquinados mostrados en la figura 54.

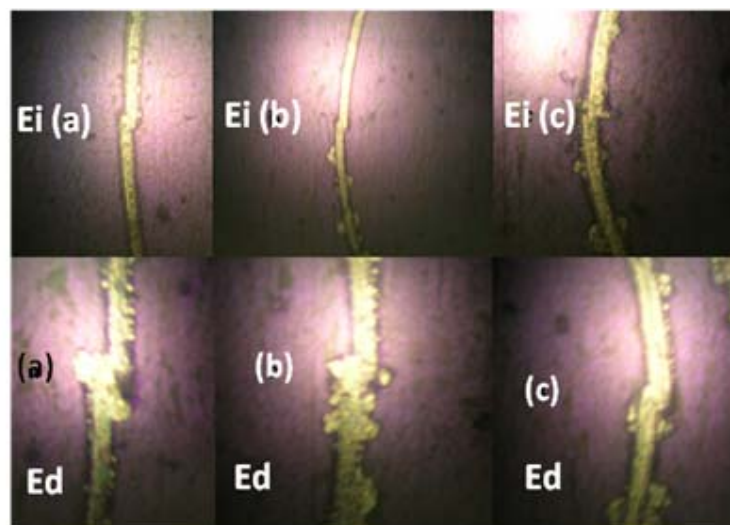


Figura 55. Errores en la compensación del juego mecánico (Extremos izquierdo y derecho).

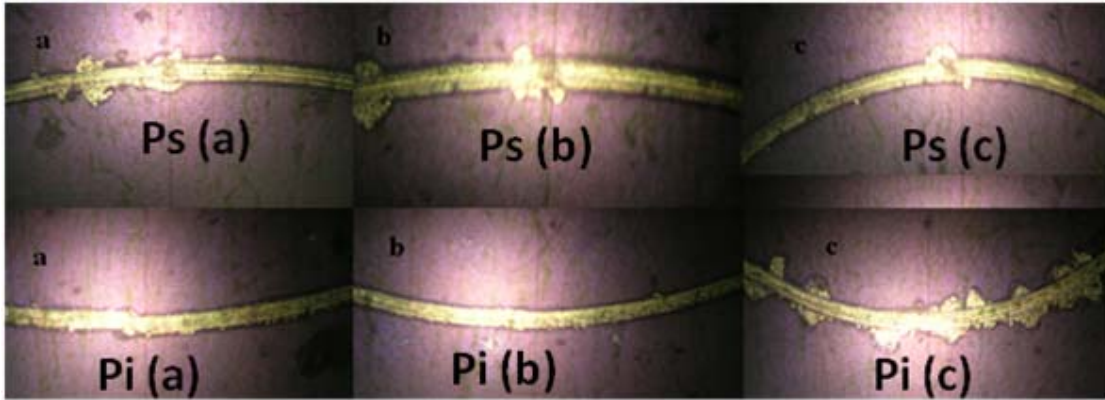


Figura 56. Errores en la compensación del juego mecánico (Polos superior e inferior).

Con el fin de obtener un error promedio de las mediciones realizadas, se analizaron estadísticamente los diámetros registrados en la tabla 15 mediante la expresión (5.4). La desviación estándar (σ) proporciona información sobre un 62.8% de la muestra, por lo que es necesario multiplicarlo por tres para tener el error en un porcentaje del 99.3% de la muestra [mathWorld, 1997].

$$\sigma = \sqrt{\frac{1}{N-1} \sum_{i=1}^N (x_i - \bar{X})^2} \dots\dots\dots (5.4)$$

Por lo tanto, los errores de estas mediciones son:

$$3\sigma_a = 0.105\mu\text{m} \Rightarrow \text{Error}_{(a)} = \pm 105\text{nm}$$

$$3\sigma_b = 0.066\mu\text{m} \Rightarrow \text{Error}_{(b)} = \pm 66\text{nm}$$

$$3\sigma_c = 0.060\mu\text{m} \Rightarrow \text{Error}_{(c)} = \pm 60\text{nm}$$

Este error nos dice cuanto se alejan, en promedio, los datos respecto de su media aritmética, y nos da una idea de que tan repetible es una circunferencia de estas dimensiones en esta zona del área de trabajo. En cuanto a lo concéntrico de los círculos, se aprecia una buena simetría entre las circunferencias.

Así pues, con los resultados obtenidos hasta el momento, podemos deducir que el juego mecánico, aún existente en el desarrollo de estas geometrías, es debido a la variación de éste mismo parámetro dentro de las regiones propuestas en la figura 52.

Conservando las magnitudes de compensación del juego mecánico hasta el momento obtenidas, se procedió a desarrollar geometrías que requirieran un mayor número de instrucciones. Estas figuras son: el escudo de los pumas y el logotipo del GMM. La razón para desarrollar estas figuras es por dos motivos, el primero es que éstas permiten ser comparadas visualmente con los resultados del sistema de control anterior, y el segundo se debe al lugar donde se desarrolló este trabajo.

5.4 Ejecución de un programa en código estándar G y M.

El procedimiento realizado para la elaboración de las geometrías propuestas fue el siguiente:

1. Dibujar ambas figura en *software* CAD.
2. Generar el proceso de manufactura en una plataforma CAM.
3. Generar el código G y M.
4. Depurar en forma manual el código obtenido en el punto anterior.
5. Colocar el código depurado en la ventanilla correspondiente de la interfaz computacional desarrollada.
6. Ejecutarlo línea por línea.
7. Representar gráficamente del progreso de manufactura.

El desarrollo de cada uno de los puntos anteriores se describe a continuación para cada una de las figuras propuestas.

5.4.1 Dibujos en CAD.

La herramienta utilizada para este propósito fue *Solid Works*, la cual resulta ser una plataforma para diseño mecánico asistido por computadora. En la figura 57 se muestra el escudo puma a maquinar. Esta figura se maquinó en tres diferentes dimensiones, la primera geometría tiene 4.86mm en el eje X y 4.96mm en el eje Y. Así pues, la segunda geometría del puma tiene 1.17mm en el eje X y 1.19mm en el eje Y. Por último, el escudo puma de dimensiones inferiores al milímetro de 970 μ m de longitud en el eje X, mientras que en el eje Y tiene 990 μ m.

Cabe mencionar, que una vez realizado el dibujo en la plataforma CAD, es necesario guardar el diseño con extensión *.xt (*Parasolid*). Esto nos permitirá exportarlo a la plataforma CAM empleada.

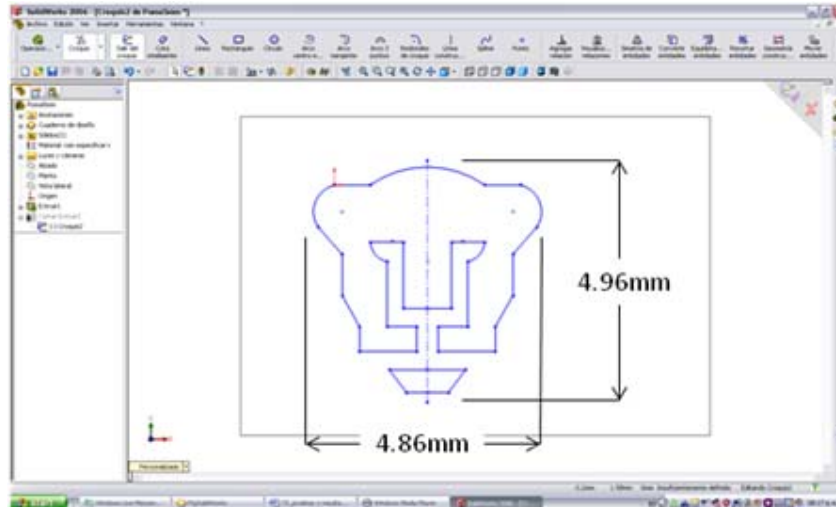


Figura 57. Escudo Puma hecho en *Solid Works*.

5.4.2 Proceso de manufactura.

La herramienta *CAM* empleada para generar el proceso de manufactura fue *VisiCAM*. En la figura 58 se muestra el resultado de la simulación del proceso de manufactura correspondiente a la geometría mostrada en la figura 57.

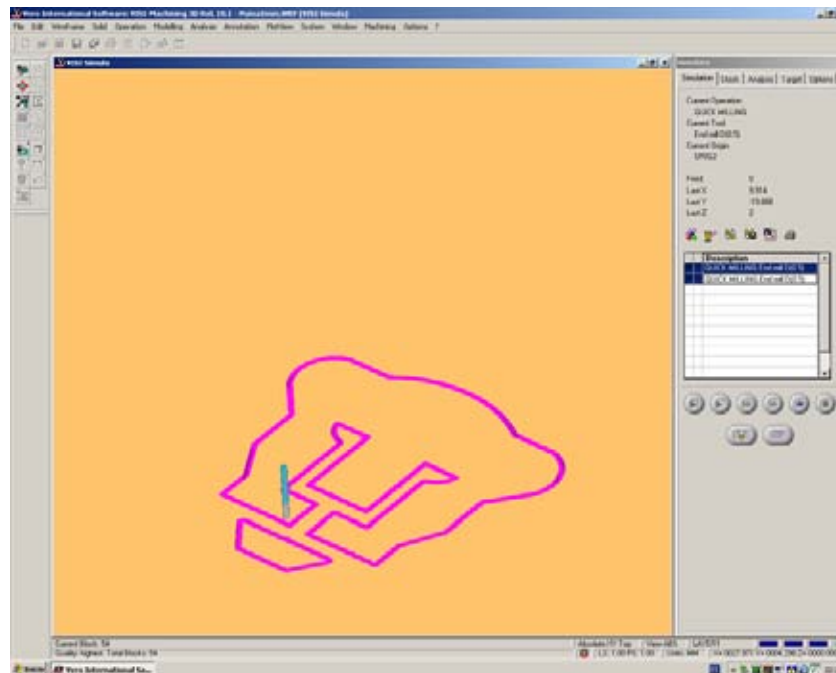


Figura 58. Proceso de manufactura del Puma en *VisiCAM*.

Una vez simulado el proceso de manufactura, se puede generar el código G y M correspondiente a dicho proceso, tal como se muestra en la figura 59, y a su vez se puede guardar con extensión *.NC, la cual permite abrir el archivo como cualquier archivo de texto (*.txt).

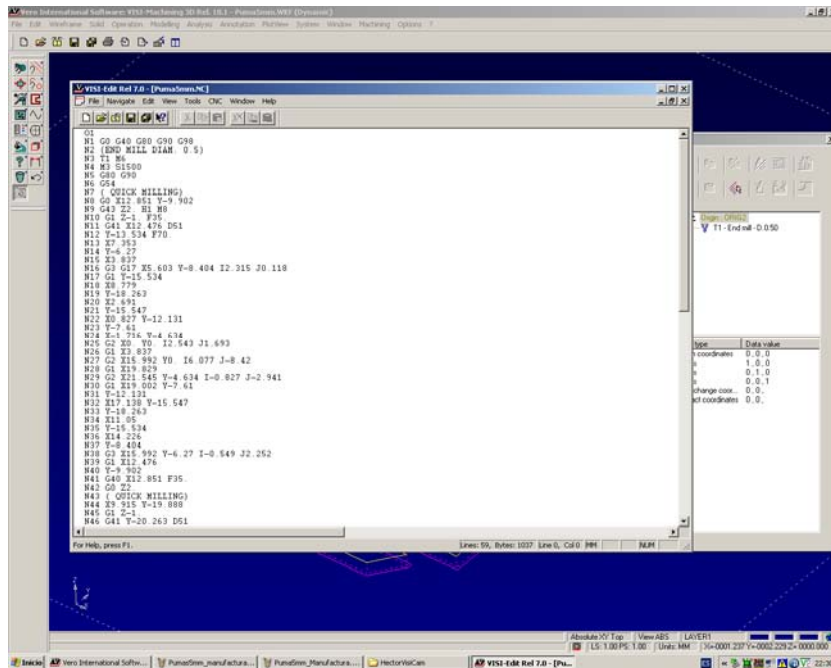


Figura 59. Código G y M generado en VisiCAM.

Una vez generado el código de manufactura, se depura con el fin de ajustar la sintaxis y eliminar aquellas instrucciones que no están especificadas en el capítulo cuatro, y que en consecuencia, no será capaz de reconocer el sistema de control implementado.

5.4.3 Ejecución del código G y M mediante la interfaz.

Teniendo el código G y M depurado para su ejecución en la interfaz descrita en el capítulo cuatro, se procede a copiarlo dentro de la caja de texto denominada *Programa CNC*, para después comenzar a ejecutar línea por línea. Esta interfaz permite al usuario verificar el desarrollo del proceso de manufactura de piezas con dimensiones no mayores a los 1.3mm. Esto es debido a que dicho proceso de manufactura se va haciendo paso por paso del actuador. En la figura 60 se muestra una pantalla de la interfaz correspondiente al proceso de maquinado del logo de los pumas de 1.2mm.

En la figura 61 se muestran dos imágenes de dimensiones similares como resultado del maquinado del escudo de los pumas. A la izquierda se muestra el maquinado de un puma de 4mm, realizado con *la interfaz computacional para el control de Micromáquinas-Herramienta* [6]. En la imagen de la derecha tenemos el maquinado del puma de 4.86mmX4.96mm, que se llevó a cabo con el sistema de control propuesto en este trabajo. Además, en la figura 62 se muestra una fotografía referente a uno de los procesos de maquinado de esta geometría; y en la tabla 16 se muestran los resultados cuantitativos de las dimensiones, así como el error relativo de partes de la figura 61 derecha calculado mediante la expresión (5.5).

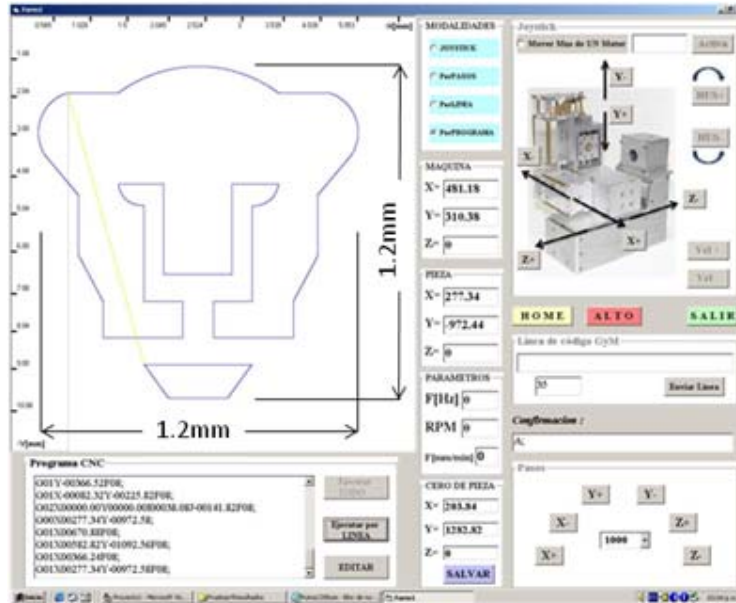


Figura 60. Pantalla del proceso de maquinado de un logo de los Pumas.

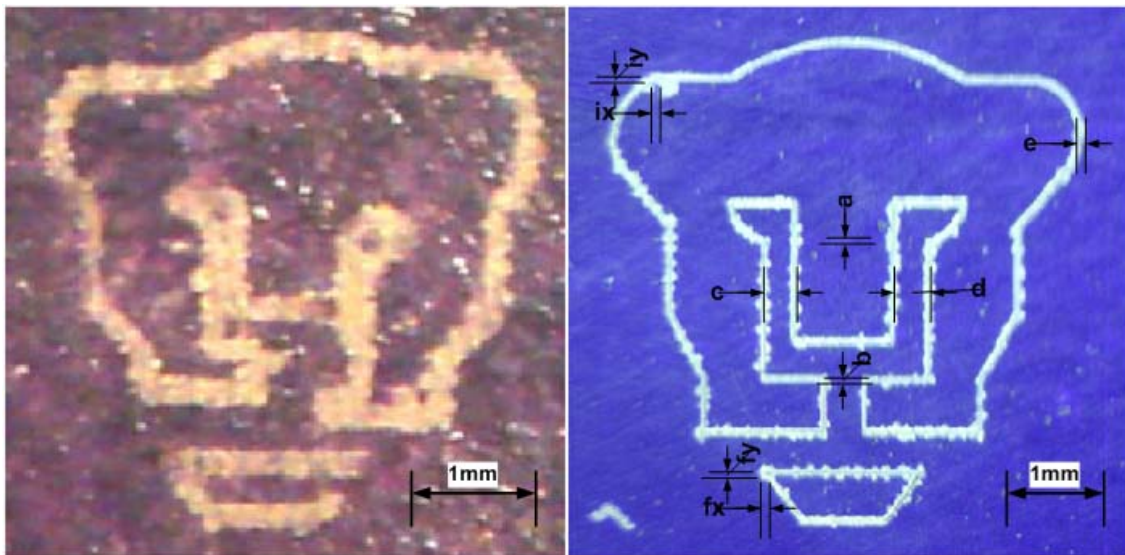


Figura 61. Puma maquinado con interfaz computacional [6] (izq), y Puma maquinado con sistema de control mediante Microcontrolador PIC (der).

$$\%E_{rel} = \frac{V_r - V_m}{V_r} \times 100 \dots\dots\dots (5.5)$$

donde: V_m : Valor medido y V_r : Valor real ó teórico.

Las magnitudes a , b , e , i_x , i_y , f_x y f_y de la figura 61 derecha, representan los errores ocasionados por la compensación hecha al juego mecánico durante el maquinado de la geometría propuesta. El error relativo obtenido representa el porcentaje de desviación del valor medido respecto al valor real de la geometría.

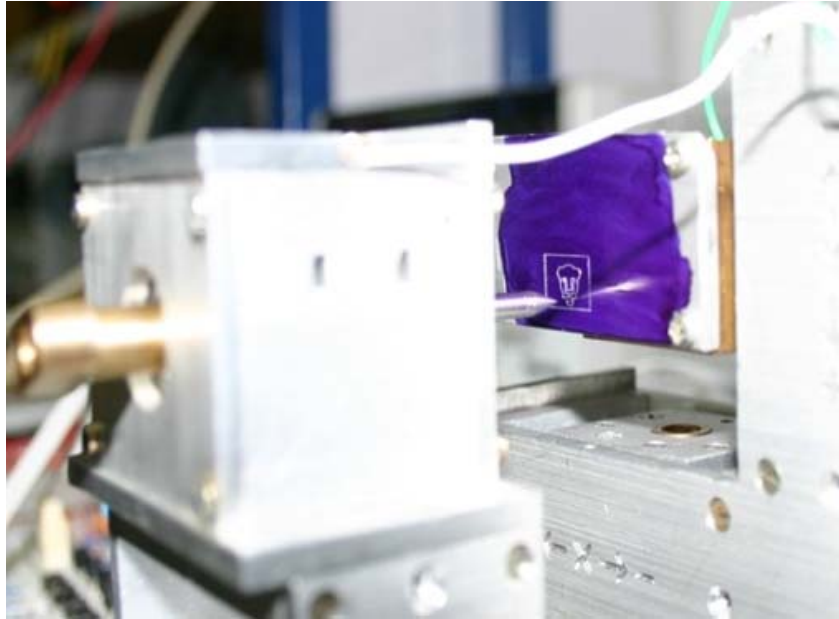


Figura 62. Proceso de maquinado del logo de los Pumas de 5mm.

Mediciones del logo de los pumas	Dimensiones teóricas [mm]	Dimensiones del maquinado [mm]	Juego mecánico [μ m]	Error relativo [%]
Longitud en X	4.86	4.830	---	0.61
Longitud en Y	4.96	4.947	---	0.26
a	---	---	Y = 11	---
b	---	---	Y = 20	---
c	X = 350×10^{-3}	X _c = 330×10^{-3}	X = 20	5.71
d	X = 350×10^{-3}	X _d = 370×10^{-3}	X = 20	5.71
e	---	---	X = 15	---
i	---	---	i _x = 12 i _y = 10	---
f	---	---	f _x = 15 f _y = 10	---

Tabla 16. Dimensiones, juego mecánico y error relativo del Puma de 5mm.

Por otro lado, en la figura 63 se muestran dos maquinados correspondientes al puma de 1.17mmX1.19mm. En el puma de la izquierda, el maquinado comienza en el punto denotado como **I** (cero de pieza) para terminar en el punto **F**. Mientras que el puma mostrado a la derecha, se maquinó en espejo, es decir, maquinando primero el lado derecho a partir del punto **Ie** para terminar en el punto **Fe** y comenzar con el lado izquierdo a partir del mismo punto **Ie**. El objetivo de cambiar el orden de maquinado de la figura 63 derecha, respecto de la izquierda, es mostrar que se pueden minimizar los errores ocasionados por la compensación del juego mecánico que aún existe en nuestro sistema.



Figura 63. Maquinado continuo de un Puma de 1.2mm (izq), y Puma maquinado en espejo de 1.2mm (der).

En la tabla 17 se aprecian las magnitudes g , h , j e Ie registradas de la figura 63 derecha. En esta tabla se observa que la diferencia entre las magnitudes h y j solo es de $3\mu\text{m}$, menor que la magnitud de la figura de la izquierda de un valor de $59\mu\text{m}$. En la figura 64 se muestra una foto del proceso de maquinado de la figura 63 derecha.

Mediciones del logo de los pumas	Dimensiones teóricas [mm]	Dimensiones del maquinado [mm]	Juego mecánico [μm]	Error relativo [%]
Longitud en X	1.17	1.165	---	0.42
Longitud en Y	1.19	1.2	---	0.84
Ie	---	---	Y = 29	---
ñ	---	---	Y = 2	---
ll	---	---	Y = 24	---
g	---	---	Y = 3	---
h	X = 80×10^{-3}	X _h = 100×10^{-3}	X = 20	25
j	X = 80×10^{-3}	X _j = 97×10^{-3}	X = 17	21.2

Tabla 17. Dimensiones y juego Mecánico del Puma de 1.2mm.

Por último, en la figura 65 se muestran dos maquinados de un puma de dimensiones inferiores al milímetro: $970\mu\text{m} \times 990\mu\text{m}$. El puma que se muestra a la izquierda fue maquinado en la parte central superior del área de trabajo, mientras que el puma de la derecha se maquinó circunscrito en la circunferencia c de la figura 54, tal como se muestra en la figura 66, con el objetivo de verificar la calidad de maquinados en espacios reducidos.

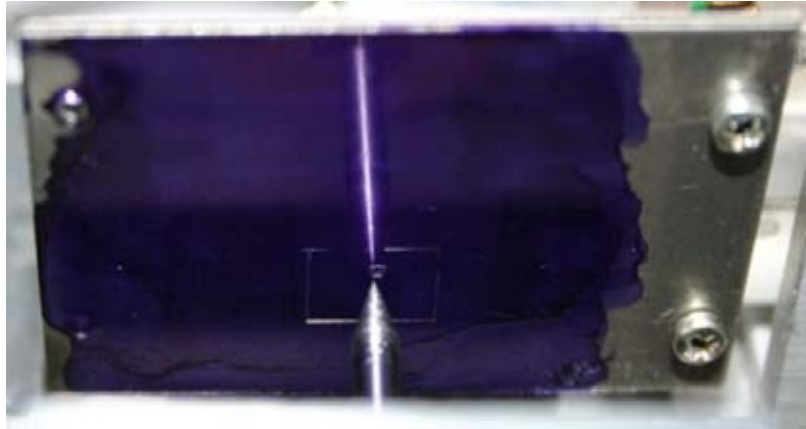


Figura 64. Proceso de maquinado del Puma de 1.2mm.

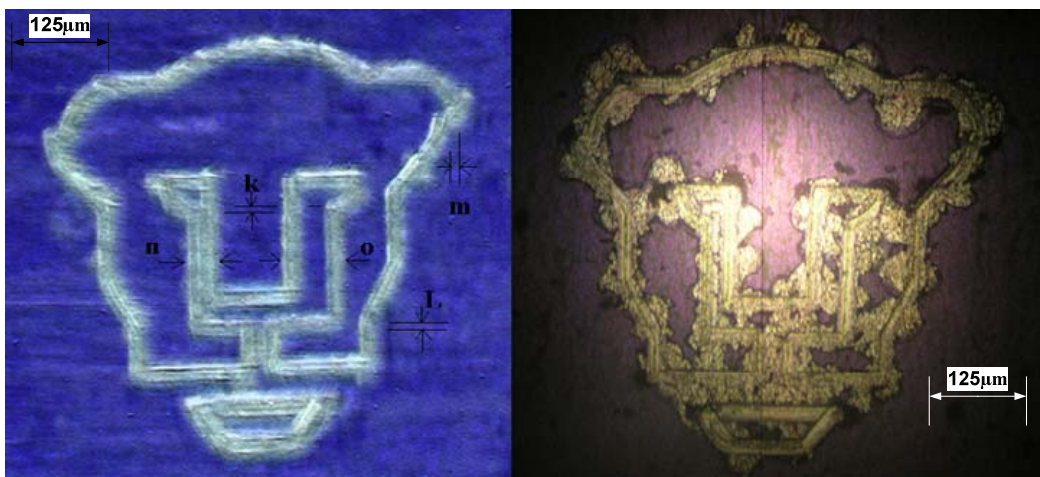


Figura 65. Dos maquinados de un Puma de 1mm.

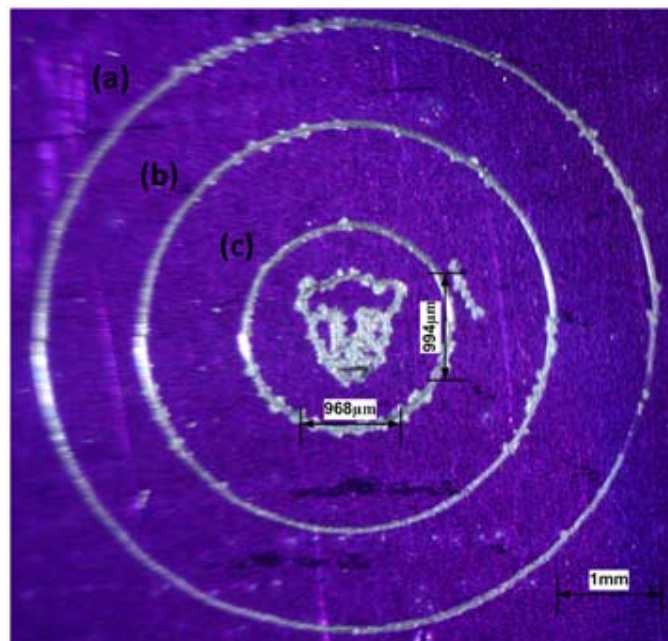


Figura 66. Puma circunscrito en circunferencia de 1mm de diámetro.

Además, en la tabla 18 se muestran los resultados cuantitativos de las dimensiones marcadas en la figura 65 izquierda. Estas magnitudes dejan ver que los errores en la compensación del juego mecánico afectan de manera considerable a las geometrías con dimensiones inferiores al milímetro, resultando una enorme desventaja para el maquinado de micropiezas.

Mediciones del logo de los pumas	Dimensiones teóricas [μm]	Dimensiones del maquinado [μm]	Juego mecánico [μm]	Error relativo [%]
Longitud en X	970	968	---	0.2
Longitud en Y	990	994	---	0.4
k	---	---	Y = 6	---
L	---	---	Y = 5	---
m	---	---	X = 29	---
n	X = 70	X _n = 45	X = 25	35.7
o	X = 70	X _o = 103	X = 33	47.1

Tabla 18. Dimensiones y Juego Mecánico del Puma de 1mm.

La segunda geometría que se desarrolló en estas pruebas, es el logotipo del GMM. Aplicando la misma metodología que en el caso del puma, se presentan en las figuras 67, 68 y 69, las imágenes referentes a los procesos de *Dibujo en CAD*, *Proceso de manufactura en VisiCAM* y *Pantalla del proceso de maquinado*, respectivamente. Las dimensiones del logotipo mostrado en la figura 67 son de 5mm de longitud en el eje X y 4.997mm en el eje Y.

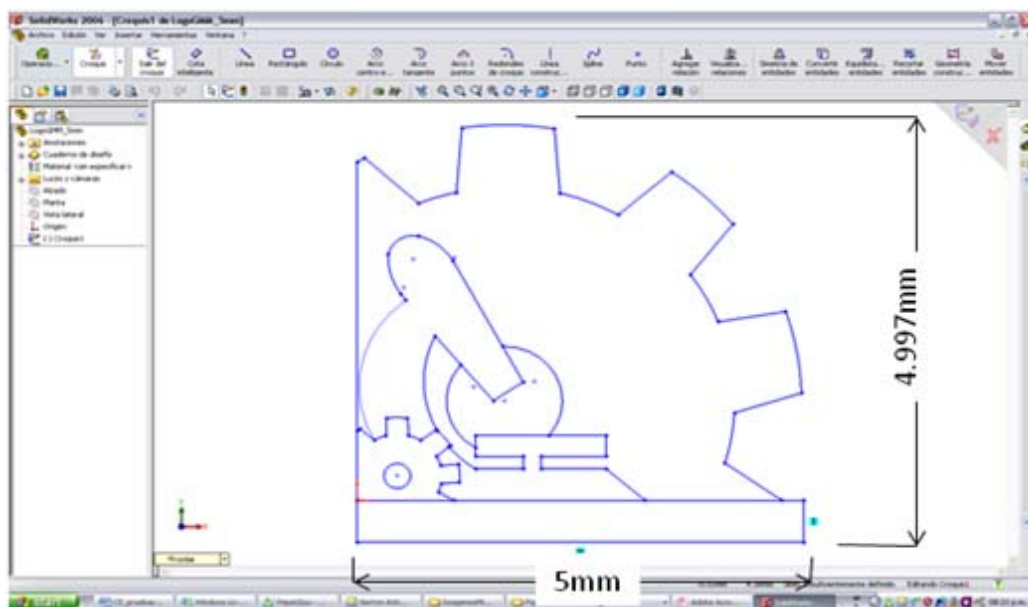


Figura 67. Logotipo del GMM hecho en Solid Works.

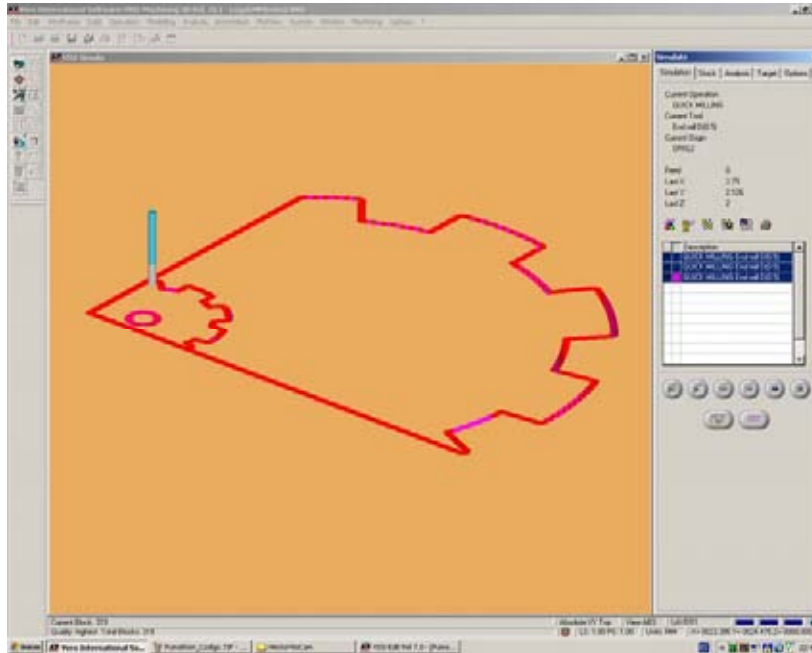


Figura 68. Proceso de manufactura del logotipo del GMM en VisiCAM.

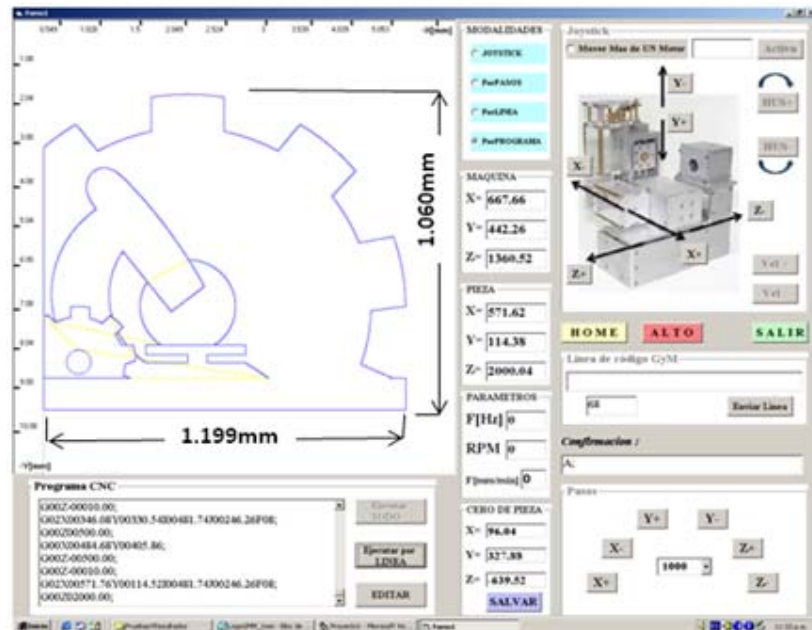


Figura 69. Pantalla del proceso de maquinado del logotipo del GMM.

Como resultado de estos procesos se tiene el maquinado mostrado en la figura 70 (derecha). A la izquierda de esta figura se muestra el maquinado del logo del GMM que se obtuvo con el *Desarrollo de una interfaz computacional para el control de Micromáquinas-Herramienta*, cuya dimensión es de 12mm. Mientras que a la derecha tenemos el maquinado que se llevo a cabo con la propuesta de este trabajo, con inicio en **i** y fin en **f**.

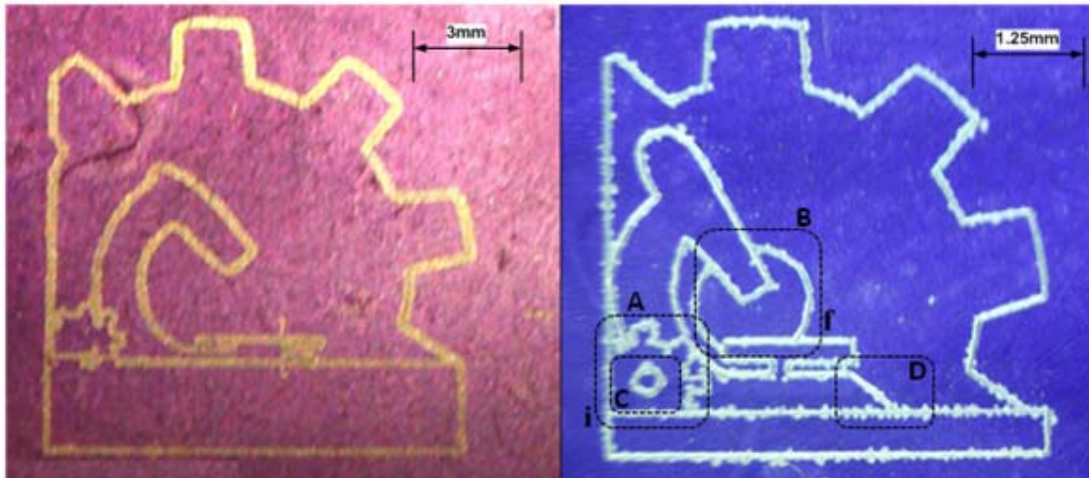


Figura 70. Dos maquinados del logotipo del GMM de 12mm (izq) y 5mm (der).

En la figura 71 se pueden apreciar algunos acercamientos del maquinado derecho de la figura 70. Estos acercamientos corresponden a las regiones A, B, C y D de la figura 70, y dentro de estas regiones se etiquetan algunas de las imperfecciones en el proceso de maquinado, cuyas magnitudes están registradas en la tabla 19.

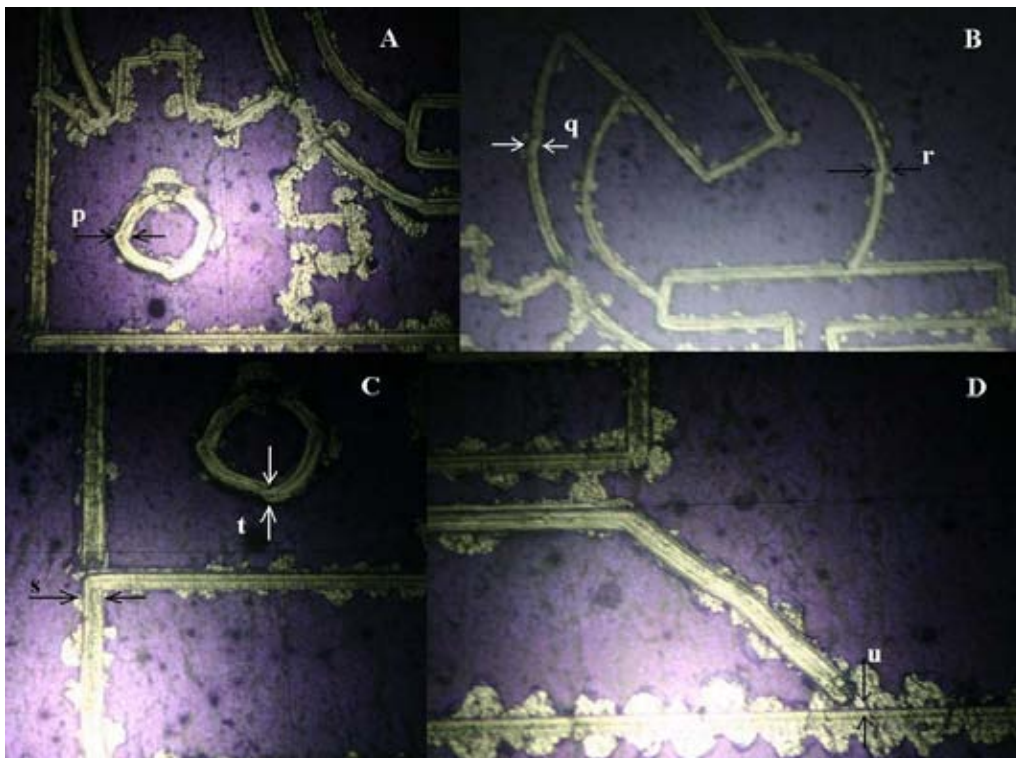


Figura 71. Acercamientos del maquinado de la figura 70 (derecha).

Mediciones del logo del GMM	Dimensiones teóricas [mm]	Dimensiones del maquinado [mm]	Juego mecánico [μm]	Error relativo [%]
Longitud en X	5	4.980	---	0.4
Longitud en Y	4.997	5.040	---	0.86
p	---	---	X = 3	---
q	---	---	X = 3	---
r	---	---	X = 4	---
s	---	---	X = 4	---
t	---	---	Y = 2	---
u	---	---	Y = 3	---

Tabla 19. Mediciones realizadas al logotipo del GMM de 5mm.

Algunas observaciones acerca del maquinado de los dos logotipos en la figura 70 son: se aprecia un mejor trazado del engrane de la región A del logotipo de la derecha, mientras que en la circunferencia de $150\mu\text{m}$ de radio de la región C se aprecian las imperfecciones debido a las micras de más por compensación al juego mecánico en las fronteras de cada cuadrante. Por otro lado, en la circunferencia de la región B se aprecia, al igual que en el caso anterior, los efectos de la compensación al juego mecánico con una repercusión menor en esta circunferencia, en cuanto a deformación, debido a las magnitud de la misma.

Para concluir con esta etapa de pruebas, se llevó a cabo el maquinado del mismo logotipo de la figura 70, cuyas dimensiones son de 1.21mm de longitud en el eje X y 1.08mm en el eje Y. El maquinado resultante se muestra en la figura 72. Además, en la tabla 20 se registran únicamente tres mediciones de la imagen de la figura 72 debido a la mala calidad del sistema óptico.

Observando la figura 72, se aprecia que son pocos los elementos que se distinguen con relativa facilidad, algunos de ellos son: el engrane de mayores dimensiones y el microscopio, y de manera muy burda el orificio del engrane externo que se etiqueto con w . Estos defectos en el maquinado son debido al número de trazos que originan cambios de dirección para llevar a cabo este logotipo, esto conlleva a realizar demasiadas compensaciones al juego mecánico, 45 para esta geometría, hecho que no pasó con el logotipo de los pumas en el cual se efectuaron entre 15 y 16 compensaciones.

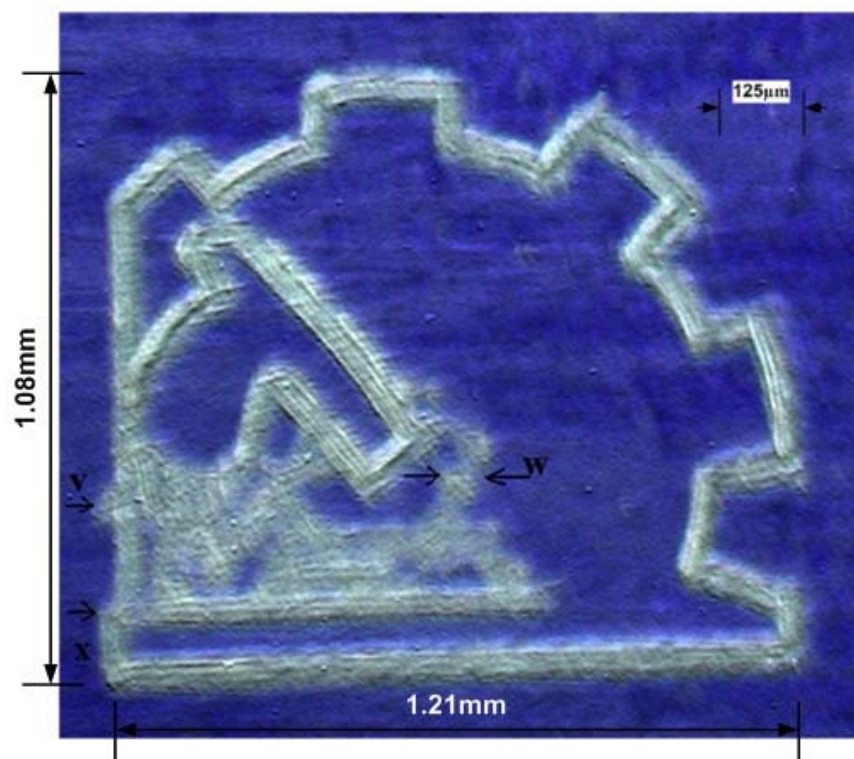


Figura 72. Maquinado del logotipo del GMM de 1mm.

Mediciones del logo del GMM	Dimensiones teóricas [mm]	Dimensiones del maquinado [mm]	Juego mecánico [µm]	Error relativo [%]
Longitud en X	1.21	1.199	---	0.9
Longitud en Y	1.08	1.060	---	1.85
v	---	---	X = 27	---
w	---	---	X = 29	---
x	---	---	X = 25	---

Tabla 20. Mediciones realizadas al logotipo del GMM de 1mm.

En los resultados mostrados en la tabla 20, se aprecia que en el maquinado de la figura 72 se obtuvo un mayor error relativo que en cualquiera de los maquinados realizados en este trabajo, superando el 1.8% respecto al diseño elaborado en CAD.

En el siguiente apartado, se establecen las conclusiones de este trabajo con base en los resultados obtenidos, tomando como referencia las mediciones realizadas y registradas en cada una de las tablas.

Conclusiones.

Al finalizar este trabajo, se concluye lo siguiente: Se analizaron y evaluaron tres algoritmos de interpolación (DDA, Escalera y DSM) para el desarrollo de geometrías con trayectorias rectas y circulares. Los métodos de interpolación seleccionados fueron: el algoritmo por DDA para segmentos rectilíneos, y el algoritmo por búsqueda directa (DSM) para trayectorias circulares, ya que éste último resultó el de menor número de iteraciones para desarrollar circunferencias, menor error radial teórico (70nm), mejor simulación, y aunque el tiempo de cada iteración es 2.7 veces el tiempo ejecutado por el método de escalera (3.2 μ s), compensa este defecto con el número de iteraciones realizadas.

El sistema de control implementado interpreta un total de 13 códigos para CNC, 8 códigos G relativos a tareas tales como: movimientos rápidos (*G00* y *G28*), interpolación lineal y circular (*G01* y *G02*, *G03*), temporizaciones (*G04*) y referencias (*G54* y *G90*). Mientras que los códigos M fueron 5, relativos a: giro del husillo (*M03* y *M04*), paro del husillo (*M05*), cambio de herramienta (*M06*) y fin de programa (*M30*). Estos comandos permiten operar una MMH de primera generación en modalidad de fresadora. Además, se establecieron de manera concreta los antecedentes respecto al campo de la Micromecánica y Sistemas CNC.

También se justificó la tendencia a implementar gradualmente características propias de sistemas CNC a sistemas de control desarrollados por el GMM, esto con el fin de simplificar y mejorar sus métodos de programación para la manufactura de micropiezas.

Se llevó a cabo un método de comparación y selección (*matriz de decisión*), para la implementación del algoritmo de interpolación y circuito de potencia a emplear. El circuito de potencia seleccionado fue el PBL3717A debido a sus características eléctricas de 1.2A en un intervalo de voltaje de los 10 a los 50V, además de que el equipo necesario para su implementación estaba dentro de los recursos del GMM. El consumo de corriente al energizar el sistema a 8V fue de 0.9A, mientras que el consumo registrado a los 12V fue de 1.3A.

Por otra parte, se propusieron e implementaron soluciones a ciertos inconvenientes que presenta el sistema de control basado en *la interfaz computacional para el control de Micromáquinas-Herramienta* [6], como lo son: las interrupciones generadas por el sistema operativo del equipo de cómputo empleado. Para este inconveniente se propuso el uso de un dispositivo de control externos a la PC, siendo propiamente el Microcontrolador PIC18F8720, del cual se utilizaron 16 pines de E/S repartidos en 3 puertos (D, E y J), un módulo USART para la comunicación serial, una interrupción para recepción de datos y 47Kb de memoria programa. Con esto también se da paso a controlar más de cuatro grados de libertad sin la necesidad de emplear otro equipo de cómputo.

También se limitó la PC a tareas de monitoreo y envío de cadenas de control. Se mejoró la simulación del proceso de manufactura en una interfaz programada, logrando que ésta representación gráfica sea paso a paso del motor, por lo que se generó una mejor aproximación al proceso de maquinado.

Se programó una interfaz en *VisualBasic* que permite operar la MMH en cuatro modos: por *joystick*, por pasos, por línea de código, y por programa CNC. Además, esta interfaz registra tres tipos de coordenadas: unas relativas al cero de máquina, otras al cero de pieza y las coordenadas del cero de pieza. También indica el avance, las RPM y la frecuencia a la que operan los actuadores de la MMH. Esto último lo hace de acuerdo a una previa caracterización del sistema, ya que como se pudo apreciar durante el desarrollo de este trabajo, no se cuenta con elementos de realimentación para cerrar un lazo de control.

Se realizaron cuatro tipos de pruebas: las primeras fueron al *hardware* electrónico, enseguida se realizaron maquinados de figuras simples para la compensación del juego mecánico, después maquinados con interpolaciones circulares, y por último la ejecución de programas en código estándar G y M. En cuanto a las pruebas realizadas al *Hardware* electrónico, se registró un consumo de 1.5A (de todo el sistema) a 14V con 48°C después de 40min de operación. Además, las señales generadas por el sistema de potencia mejoraron al implementar un circuito LC en cada fase de un motor de la MMH.

Con el maquinado de figuras simples se obtuvo una magnitud promedio de juego mecánico por compensar en el eje X de 204 μ m, mientras que en el eje Y resultaron 178 μ m. Por otro lado, en los maquinados con interpolaciones en dos ejes (circunferencias), se concluye un error promedio de ± 77 nm. Además, las magnitudes E y P de las circunferencias (*a*, *b* y *c*) maquinadas demuestran que aún existen micras por compensar, de lo cual se concluye que el juego mecánico varía en toda el área de trabajo. El error correspondiente a estas micras por compensar es de $\pm 33\mu$ m en el eje X, y $\pm 27\mu$ m en el eje Y.

De la ejecución de programas en código estándar, se concluyen buenos resultados al comparar los maquinados del logo de los Pumas y del logo del GMM, hechos por el sistema de control basado en la *interfaz computacional* y los desarrollados por este trabajo, teniendo un error relativo máximo en el eje X de 0.9% y 1.86% en el eje Y respecto a sus medidas de diseño en CAD. Sin embargo, es importante aclarar que el tiempo requerido para llevar a cabo los maquinados con la propuesta de este trabajo resultó mayor a la del sistema de control anterior, y esto fue debido a problemas referidos al canal de comunicación empleado, que restringió la velocidad de los motores a 166 [Hz].

Por último, con este sistema de control se demostró que el maquinado en espejo de las figuras (que su geometría permita) puede minimizar los errores ocasionados por el juego mecánico. Además, se maquinaron geometrías conformadas por trazos de dimensiones menores al milímetro (logo puma de 1mm), con una muy buena aproximación, lo cual representa un gran avance para el GMM. Sin embargo, al aumentar el número de cambios de dirección en la geometría (logo GMM 1mm), ésta se ve afectada considerablemente por la variación del juego mecánico en toda el área de trabajo. En conclusión, el sistema de control tendrá problemas para maquinar geometrías de dimensiones inferiores al milímetro, si es que el número de compensaciones para maquinar es alto (>16), y que debido a su geometría no permita maquinarla en espejo.

TRABAJO A FUTURO.

Con el fin de mejorar el proceso de manufactura de micropiezas en el GMM, a nivel de elementos mecánicos, se pretenden hacer algunas modificaciones respecto a los tornillos sin fin, el tipo de rodamientos empleados, características de los motores utilizados, etc., esperando con esto reducir de cierta forma el juego mecánico de sistema. A nivel de *hardware* electrónico, se espera poder implementar etapas de potencia basadas en el circuito A3988, descrito en este trabajo, con el fin de reducir espacios y mejorar el estado de las señales producidas para el control de los actuadores.

En cuanto a la tarjeta controladora, se pretende implementar un control distribuido, es decir, utilizar una topología de varios Microcontroladores PIC (gobernados por un maestro) y bloques de memoria externos para el almacenamiento del programa pieza, permitiendo esto delegar funciones al resto del sistema. Con lo anterior, será posible incorporar ciclos enlatados, ya que estos combinan una gran cantidad de operaciones de programación estándar y disminuyen los pasos de programación, simplificando las operaciones matemáticas y optimizando las condiciones de corte.

Por otro lado, es necesario implementar otro protocolo de comunicación, por ejemplo, el basado en USB (*Universal Serial Bus*) con el fin de incrementar la velocidad en la transmisión de datos para el proceso de manufactura. También, es necesario agregar a la interfaz desarrollada características que la hagan más versátil, por ejemplo, importar el programa CNC de cualquier localidad dentro de la PC. Ejecutar el programa CNC de manera continua. Tener un proceso de depuración de código CNC, etc.

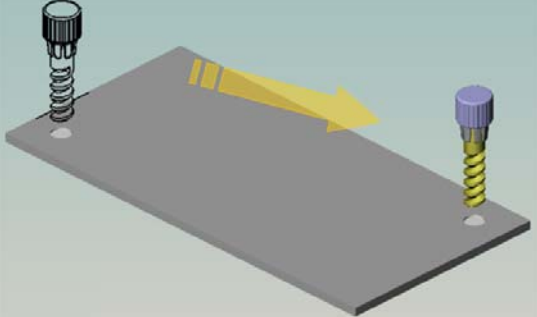
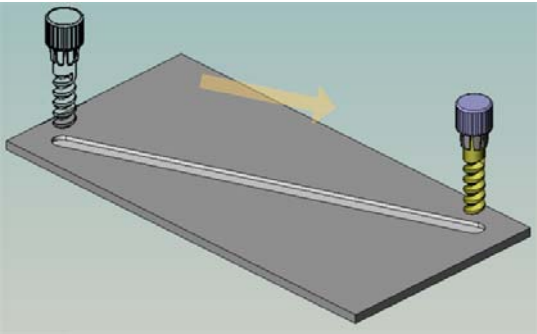
En lo que respecta a la compensación por *software* del juego mecánico, se podría hacer una división más fina del área de trabajo, con el fin de realizar una mejor compensación en las diferentes regiones de ésta. Además, sería bueno monitorear cada una de las regiones del área de trabajo, en las que se encuentra la herramienta para compensar de acuerdo a la región del área de trabajo.

Es necesario hacer que el sistema desarrollado reconozca instrucciones G y M para torno, con el fin de ampliar las aplicaciones de esta MMH. Por último, en la interpretación gráfica del proceso de manufactura en la interfaz de usuario, debe implementarse una escala, con el fin de abarcar geometrías de dimensiones similares al área de trabajo de la MMH. Además, es importante agregar elementos de realimentación para cerrar el lazo de control, logrando así obtener mayores revoluciones de los actuadores empleados, un mejor monitoreo del sistema y una mejor interpretación gráfica.

Anexos

A.1 Descripción y sintaxis de los códigos.

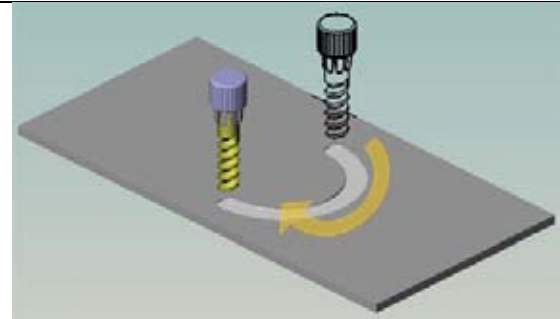
La descripción y sintaxis, de cada una de las instrucciones G y M propuestas para este sistema de control, se muestran en las tablas 23 y 24, respectivamente. Podemos observar en la primera columna, el tipo y número de código (*en negritas*) seguido de su sintaxis. En la siguiente columna se describe la tarea realizada por dicho código, mientras que, en la última columna se muestra una pequeña ilustración del proceso.

CÓDIGOS “G” y SINTAXIS	DESCRIPCIÓN	PROCESO
<p>G00X±____Y±____;</p>	<p>Este comando desplaza la herramienta al punto indicado por las coordenadas “X” y “Y” (de 5 dígitos), a la velocidad máxima de desplazamiento de la máquina y en línea recta. En su recorrido no se efectúa arranque de material.</p>	
<p>G01X±____Y±____F__;</p>	<p>El comando genera el movimiento simultáneo de uno ó dos ejes, desde su posición actual hasta alcanzar la posición final, especificada por la coordenadas “X” y “Y” (de 5 dígitos) mediante una trayectoria rectilínea, y a una velocidad de trabajo determinada por el avance “F” (de 2 dígitos).</p>	

G02X±____Y±____I±____J±____F__;

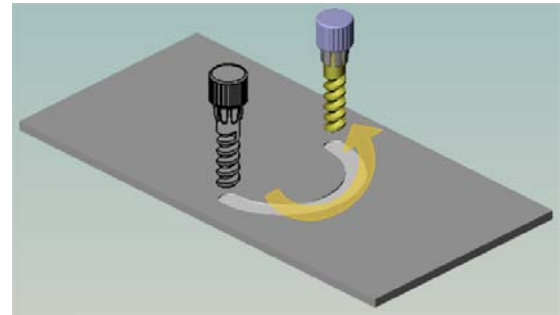
Pag. 39


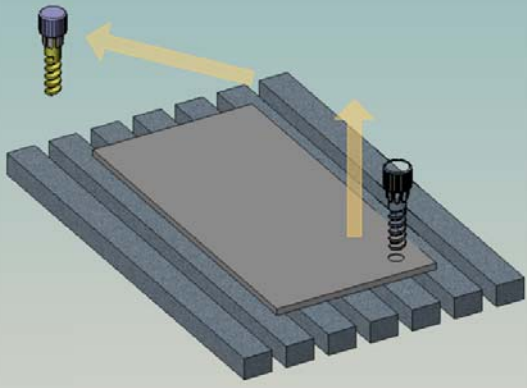

Este comando se utiliza para efectuar un movimiento en arco de circunferencia, en sentido de las manecillas del reloj, a partir del punto donde se encuentra hasta el punto final indicado por las coordenadas "X" y "Y" (de 5 dígitos), y tomando como referencia el centro del arco denotado por las coordenadas "I" y "J" (de 5 dígitos). La velocidad de desplazamiento estará sujeta al avance "F" (de 2 dígitos).



G03X±____Y±____I±____J±____F__;

Comando que al igual que G02, se utiliza para efectuar un movimiento en arco de circunferencia, en sentido contrario al de las manecillas del reloj, a partir del punto donde se encuentra hasta el punto final indicado por las coordenadas "X" y "Y" (de 5 dígitos), y tomando como referencia el centro del arco denotado por las coordenadas "I" y "J" (de 5 dígitos). La velocidad de desplazamiento estará sujeta al avance "F" (de 2 dígitos).

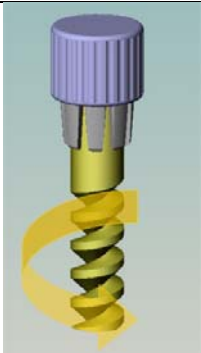
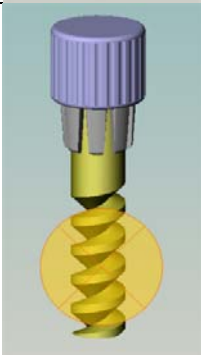
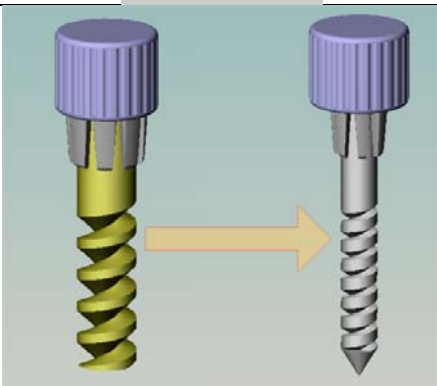


<p>G04P__;</p> <p>Pag. 40</p>	<p>Con esta instrucción se obtiene una pausa durante un tiempo especificado por “P” (en segundos y de 2 dígitos). Una vez transcurrido este tiempo la máquina continúa automática el proceso que sigue.</p>	
<p>G28X0Y0; ó G28Z0;</p>	<p>Este comando permite trasladar automáticamente la herramienta a la posición cero de máquina en el(los) eje(s) correspondiente(s). Sus movimientos describen trayectorias rectilíneas a máxima velocidad. Generalmente es empleado en los cambios de herramienta.</p>	
<p>G54;</p>	<p>Reserva una localidad de memoria en la cual se almacenará el cero de pieza, y a la cual se referirán todos los datos de posición de la pieza a manufacturar.</p>	

<p style="text-align: center;">G90;</p> <p style="text-align: center;">Pag. 41</p>	<p>Establece al sistema con dimensiones en modo absoluto. Este modo utiliza como punto de referencia el cero de pieza almacenado por G54.</p>	
---	---	--

Tabla 21. Conjunto de códigos “G” a interpretar.

CÓDIGOS “M” y SINTAXIS	DESCRIPCIÓN	PROCESO
<p style="text-align: center;">M03S___;</p>	<p>Comando que activa el giro del husillo principal en sentido horario. La velocidad del husillo, estará determinada por las revoluciones dadas en la misma sintaxis del comando (<i>spindle</i>).</p>	

<p>M04S____;</p>	<p>Este comando activa el giro del husillo principal en sentido contrario a las manecillas del reloj. La velocidad del husillo, estará determinada por las revoluciones dadas en la misma sintaxis del comando (<i>spindle</i>).</p>	
<p>M05;</p> <p>Pag. 42</p>	<p>Comando que detiene el giro del husillo. Anula las funciones M03 y M04.</p>	
<p>M06T__;</p>	<p>Procedimiento que prepara a la máquina para llevar a cabo un cambio de herramienta, ubicada en la posición identificada con el número proporcionado en la sintaxis (<i>tool</i>), del almacén del magazine de herramientas.</p>	

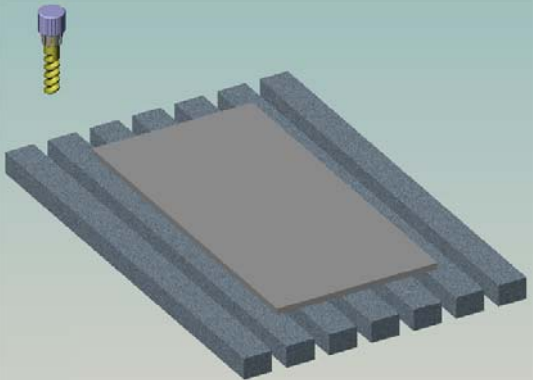
<p style="text-align: center;">M30;</p> <p style="text-align: center;">Pag. 43</p>	<p>Termina y reinicia el programa CNC de vuelta a la secuencia No.1.</p>	
---	--	---

Tabla 22. Conjunto de códigos “M” a interpretar.

A.2 Comandos para activar modos de operación en la interfaz computacional.

Modalidad de operación	Sintaxis de la cadena de control
Por joystick	JX±; (modo joystick/eje/sentido/fin de línea)
Por joystick (movimiento simultáneo de dos o más ejes)	&X±Y±Z±; (más de un eje/eje1/sentido/.../eje3/sentido/fin de línea)
Por pasos	PX±1000; (modo pasos/eje/sentido/cifra cuatro dígitos/fin de línea)
Por línea y Por programa	En este modo la cadena de control es ocupada por cualquiera de los comandos descritos en las tablas 11 y 12 de esta misma sección.
Para mandar a HOME	H; (comando de home/fin de línea)
Para activar el ALTO	A; (comando de alto/fin de línea)
Para determinar el juego mecánico	BX200; (comando para modificar el juego mecánico/eje/cifra de tres dígitos/fin de línea)

Para activar el movimiento a una frecuencia determinada.	FX±250; (comando para activar el movimiento a una frecuencia /eje/sentido/fin de línea)
Para el incremento de velocidad	W; (comando para incrementar velocidad/fin de línea)
Para el decremento de velocidad	V; (comando para disminuir velocidad/fin de línea)
Para almacenar el cero de pieza	Q; (comando para almacenar cero de pieza/fin de línea)

Tabla 23. Modos de operación de la interfaz computacional.

A.3 Código de la Interpolación líneal con DDA.

```

'***** RUTINA DE INTERPOLACIÓN LINEAL CON DDA *****
INTERLINEALX:    QX = QX + PX
                  GoTo PRIMERACONDx
INTERLINEALY:    QY = QY + PY
                  GoTo PRIMERACONDy
PRIMERACONDx:    If ContadorX >= PX Then GoTo INTERLINEALY
                  If QX > HIPO Then QX = QX - HIPO : Inc ContadorX : diopasoex = 1 : GoTo SECUENCIADDALINEAL
                  GoTo INTERLINEALY
PRIMERACONDy:    If ContadorY >= PY Then GoTo VECONTEOX
                  If QY > HIPO Then QY = QY - HIPO : Inc ContadorY : diopasoey = 1 : GoTo CHECAEJEDDALINEALY
                  If diopasoex = 1 Then GoTo AVANCEPORJOYSTICK
                  GoTo INTERLINEALX

VECONTEOX:       If diopasoex = 1 Then GoTo AVANCEPORJOYSTICK
                  If ContadorX > PX Then
                      GoTo ALTO
                  ElseIf ContadorX = PX Then
                      GoTo ALTO
                  Else
                      GoTo INTERLINEALX
                  EndIf

```

```

'***** ESTABLECIENDO LA SECUENCIA DE PASO PARA LA INTERPOLACIÓN LINEAL *****
SECUENCIADDALINEAL:  If sentidox = 0 Then GoTo ACTUALIZAEDODDALINEALX
                     If ESTADox = 1 Then
                       ESTADox = 4
                       GoTo COMBINACIONESDDALINEAL
                     Else
                       Dec ESTADox
                       GoTo COMBINACIONESDDALINEAL
                     Endif
ACTUALIZAEDODDALINEALX: If ESTADox = 4 Then
                       ESTADox = 1
                       GoTo COMBINACIONESDDALINEAL
                     Else
                       Inc ESTADox
                       GoTo COMBINACIONESDDALINEAL
                     Endif
CHECAEJEDDALINEALY:  If sentidoy = 1 Then GoTo ACTUALIZAEDODDALINEALY
                     If ESTADoy = 1 Then
                       ESTADoy = 4
                       GoTo OTROEJEDDALINEAL
                     Else
                       Dec ESTADoy
                       GoTo OTROEJEDDALINEAL
                     Endif
ACTUALIZAEDODDALINEALY: If ESTADoy = 4 Then
                       ESTADoy = 1
                       GoTo OTROEJEDDALINEAL
                     Else
                       Inc ESTADoy
                       GoTo OTROEJEDDALINEAL
                     Endif

COMBINACIONESDDALINEAL: Select ESTADox
                          Case 1
                            Set COMBINACION.0
                            Clear COMBINACION.1
                            GoTo INTERLINEALY
                          Case 2
                            Set COMBINACION.0

```

```

Set COMBINACION.1
GoTo INTERLINEALY
Case 3
Clear COMBINACION.0
Set COMBINACION.1
GoTo INTERLINEALY
Case Else
Clear COMBINACION.0
Clear COMBINACION.1
GoTo INTERLINEALY
EndSelect

```

OTROEJEDDALINEAL:

```

Select ESTADoy
Case 1
Set COMBINACION.2
Clear COMBINACION.3
GoTo AVANCEPORJOYSTICK
Case 2
Set COMBINACION.2
Set COMBINACION.3
GoTo AVANCEPORJOYSTICK
Case 3
Clear COMBINACION.2
Set COMBINACION.3
GoTo AVANCEPORJOYSTICK
Case Else
Clear COMBINACION.2
Clear COMBINACION.3
GoTo AVANCEPORJOYSTICK
EndSelect

```

A.4 Código de la Interpolación circular con DDA.

****** RUTINA DE INTERPOLACION CIRCULAR 4TO. Y 2DO. CUADRANTE ******

```

INTERCIRCULAR4: PY = RADIO
PX = 0
QX = 0

```

```

QY = 0
CONTADORy = 0
CONTADORx = 0
INTERCIRCULAR4Y: QY = QY + PY
GoTo PRIMERACOND4y
INTERCIRCULAR4X: QX = QX + PX
GoTo PRIMERACOND4x
PRIMERACOND4y: If CONTADORy >= RADIO Then GoTo INTERCIRCULAR4X
If QY > RADIO Then QY = QY - RADIO : Inc CONTADORy : DIOPASOENY = 1 : Inc PX : GoTo CHECAEJE4Y
' INCY = 1 :
GoTo INTERCIRCULAR4X
PRIMERACOND4x: If CONTADORx >= RADIO Then GoTo VECONTEO4Y
If QX > RADIO Then QX = QX - RADIO : Inc CONTADORx : DIOPASOENX = 1 : Dec PY : GoTo
SECUENCIAAMBOS4 'PRIMERACONDy 'INCX = 1 :
If DIOPASOENY = 1 Then GoTo PASOSINTERCIRCULAR4
GoTo INTERCIRCULAR4Y 'PASOSINTERLINEAL
VECONTEO4Y: If CONTADORy >= RADIO Then Inc CUADRANTE : GoTo CUIDADOCONCUAD
GoTo PASOSINTERCIRCULAR4

```

'***** RUTINA DE INTERPOLACION CIRCULAR 3ER. Y 1ER. CUADRANTE *****'

```

INTERCIRCULAR1: PX = RADIO
PY = 0
QX = 0
QY = 0
CONTADORy = 0
CONTADORx = 0
INTERCIRCULAR1X: QX = QX + PX
GoTo PRIMERACOND1x
INTERCIRCULAR1Y: QY = QY + PY
GoTo PRIMERACOND1y
PRIMERACOND1x: If CONTADORx >= RADIO Then GoTo INTERCIRCULAR1Y
If QX > RADIO Then QX = QX - RADIO : Inc CONTADORx : DIOPASOENX = 1 : Inc PY : GoTo
SECUENCIAAMBOS1 'PRIMERACONDy 'INCX = 1 :
GoTo INTERCIRCULAR1Y
PRIMERACOND1y: If CONTADORy >= RADIO Then GoTo VECONTEO1X

```



```

If QY > RADIO Then QY = QY - RADIO : Inc CONTADORy : DIOPASOENY = 1 : Dec PX : GoTo CHECAEJE1Y
'INCY = 1 :
If DIOPASOENX = 1 Then GoTo PASOSINTERCIRCULAR1
GoTo INTERCIRCULAR1X          'PASOSINTERLINEAL

VECONTEO1X: If CONTADORx >= RADIO Then Inc CUADRANTE : GoTo CUIDADOCONCUAD
GoTo PASOSINTERCIRCULAR1

```

A.5 Código de la Interpolación circular por Escalones.

```

MODOINTERCIRCULAR: If DATO = 255 Then PARARINTERCIRCULAR
If MAGNITUDRADIO = 0 Then RADIO = DATO * 1000 : MAGNITUDRADIO = 1 : retfie fast
AVANCE = DATO
RADIO = RADIO / RESOL
PPR = 1000 / RESOL
FCIAALGORITMO = (PPR * AVANCE) / 60
PDOALGORITMO = 1000000 / FCIAALGORITMO
Yo = RADIO
GoTo SELECCIONACUAD

CUIDADOCONCUAD: If CUADRANTE = 5 Then CUADRANTE = 1 : GoTo PARARINTERCIRCULAR
SELECCIONACUAD: Select CUADRANTE
Case 1
SENTIDOx = 1
SENTIDOy = 0
GoTo INTERCIRCULAR1
Case 2
SENTIDOx = 0
SENTIDOy = 0
GoTo INTERCIRCULAR4
Case 3
SENTIDOx = 0
SENTIDOy = 1
GoTo INTERCIRCULAR1
Case Else
SENTIDOx = 1
SENTIDOy = 1
GoTo INTERCIRCULAR4

```

EndSelect

*'***** RUTINA DE INTERPOLACION CIRCULAR 3ER. Y 1ER. CUADRANTE *****'*

```
INTERCIRCULAR1:  D = Xo * Xo
                  D = D + (Yo * Yo)
                  D = D - (RADIO * RADIO)
                  If Yo = 0 Then Inc CUADRANTE : GoTo CUIDADOCONCUAD
                  If D <= 0 Then Inc Xo : GoTo SECUENCIAAMBOS1      'PRIMERACONDY      'INCX = 1 :
                  Dec Yo
                  GoTo CHECAEJE1Y
```

*'***** RUTINA DE INTERPOLACION CIRCULAR 4TO. Y 2DO. CUADRANTE *****'*

```
INTERCIRCULAR4:  D = Xo * Xo
                  D = D + (Yo * Yo)
                  D = D - (RADIO * RADIO)
                  If Xo = 0 Then Inc CUADRANTE : GoTo CUIDADOCONCUAD
                  If D <= 0 Then Inc Yo : GoTo CHECAEJE4Y      'PRIMERACONDY      'INCX = 1 :
                  Dec Xo
                  GoTo SECUENCIAAMBOS4
```

A.6 Código de la Interpolación circular por DSM.

*'***** RUTINA PARA EJECUTAR LA INETRPOLACIÓN CIRCULAR CON DSM *****'*

```
DSMCIRCULARCUAD1:  DeltaX = (2 * Xo) + 1
                   DeltaY = (-2 * Yo) + 1
                   GoTo DSM
```

```
DSMCIRCULARCUAD2:  DeltaX = (-2 * Xo) + 1
                   DeltaY = (2 * Yo) + 1
```

```
DSM:              D1 = Abs(D + DeltaX)
                   D2 = Abs(D + DeltaY)
```

```

D3 = Abs(D + DeltaX + DeltaY)
If D1 > D2 Then GoTo COMPARAD2COND3
If D1 > D3 Then
    D = D + DeltaX + DeltaY
    DeltaX = DeltaX + 2
    DeltaY = DeltaY + 2
    diopasoenx = 1
    diopasoeny = 1
    GoTo ACTUALIZACONTADORES
Else
    D = D + DeltaX
    DeltaX = DeltaX + 2
    diopasoenx = 1
    diopasoeny = 0
    GoTo ACTUALIZACONTADORES
Endif
COMPARAD2COND3:
If D2 > D3 Then
    D = D + DeltaX + DeltaY
    DeltaX = DeltaX + 2
    DeltaY = DeltaY + 2
    diopasoenx = 1
    diopasoeny = 1
    GoTo ACTUALIZACONTADORES
Else
    D = D + DeltaY
    DeltaY = DeltaY + 2
    diopasoenx = 0
    diopasoeny = 1
    GoTo ACTUALIZACONTADORES
Endif
ACTUALIZACONTADORES:
Select CUADRANTE
Case 1
    If NoCodigoG = 2 Then
        If diopasoenx = 1 Then Inc Xo
        If diopasoeny = 1 Then Dec Yo
    Else
        If diopasoenx = 1 Then Dec Xo
        If diopasoeny = 1 Then Inc Yo

```

```

    EndIf
Case 2
  If NoCodigoG = 2 Then
    If diopasoenx = 1 Then Dec Xo
    If diopasoeny = 1 Then Inc Yo
  Else
    If diopasoenx = 1 Then Inc Xo
    If diopasoeny = 1 Then Dec Yo
  EndIf
Case 3
  If NoCodigoG = 2 Then
    If diopasoenx = 1 Then Inc Xo
    If diopasoeny = 1 Then Dec Yo
  Else
    If diopasoenx = 1 Then Dec Xo
    If diopasoeny = 1 Then Inc Yo
  EndIf
Case Else
  If NoCodigoG = 2 Then
    If diopasoenx = 1 Then Dec Xo
    If diopasoeny = 1 Then Inc Yo
  Else
    If diopasoenx = 1 Then Inc Xo
    If diopasoeny = 1 Then Dec Yo
  EndIf
EndSelect

If diopasoenx = 1 Then GoTo SECUENCIADSMCIRCULAR
If diopasoeny = 1 Then GoTo CHECAEJEDSMCIRCULAR
GoTo DSM

```

A.7 Diagramas y tarjetas electrónicas.

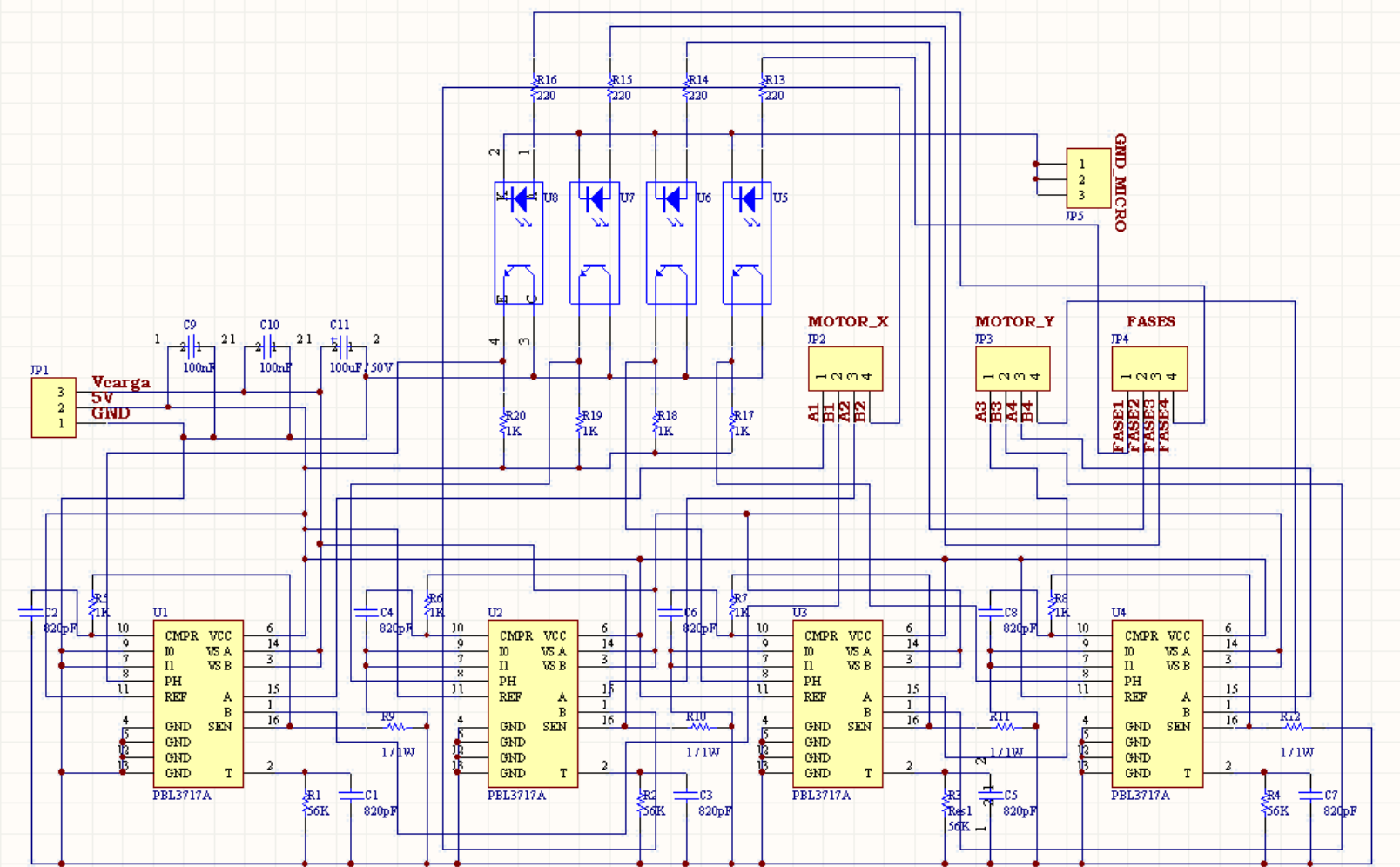


Figura 73. Esquemático del sistema de potencia para dos motores mostrado en la figura 42.

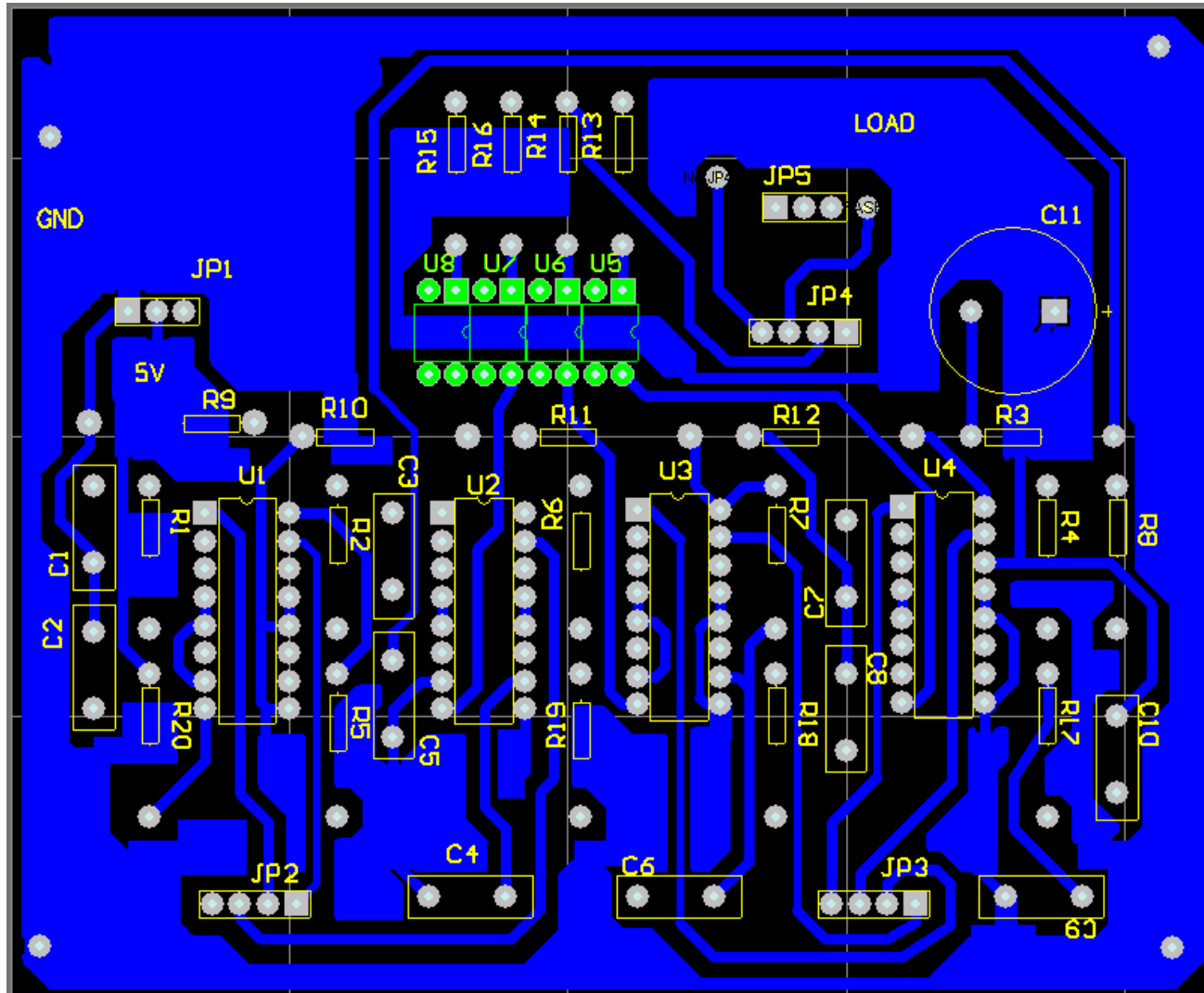


Figura 74. Cara inferior del circuito impreso para el sistema de potencia mostrado en la figura 42.

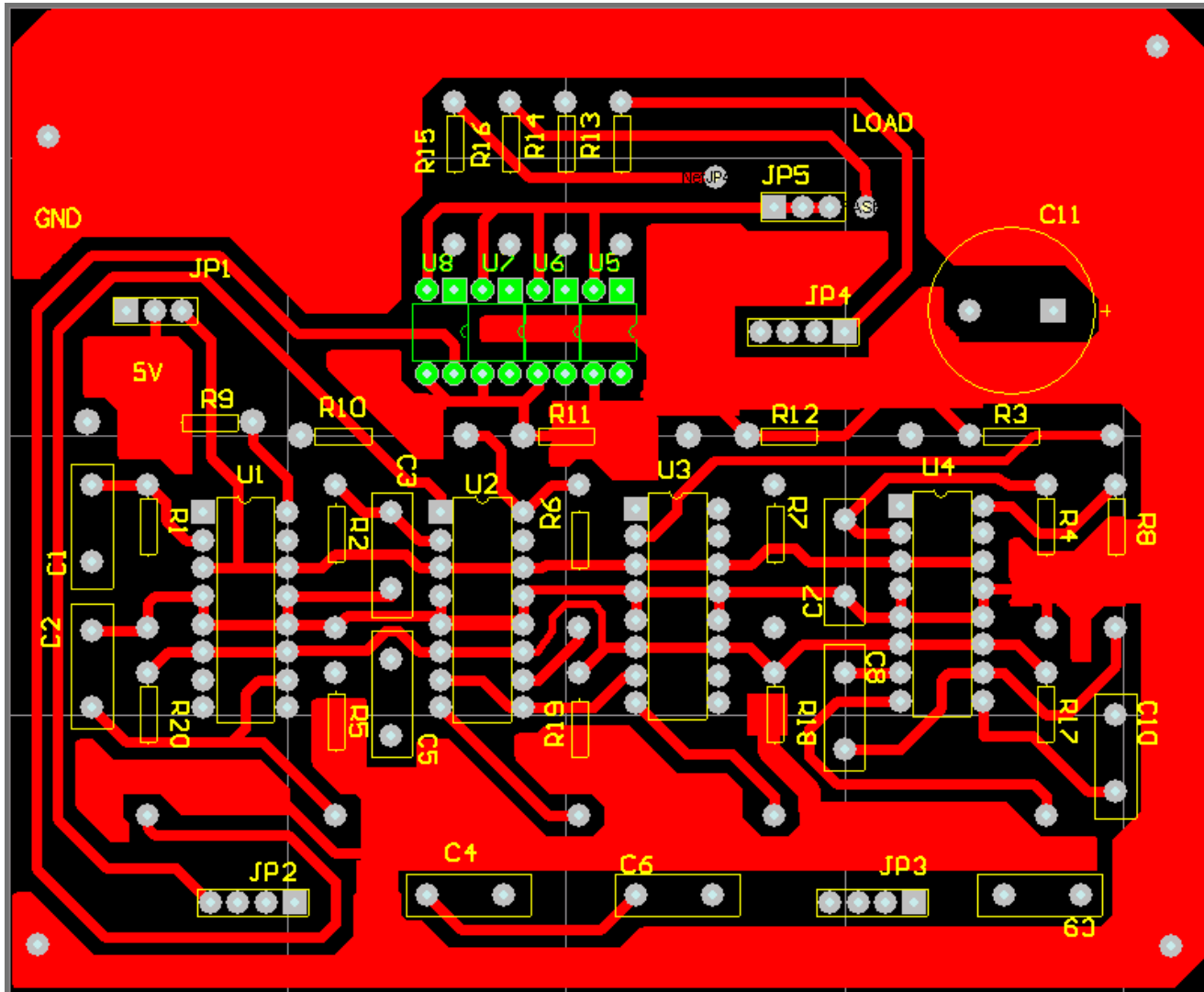


Figura 75. Cara posterior del circuito impreso para el sistema de potencia mostrado en la figura 42.

Glosario.

<i>A</i>	: Ampere.
<i>AMD</i>	: <i>Advanced Micro Devices, Inc.</i>
<i>Byte</i>	: Conjunto de 8 bits.
<i>CAD</i>	: Diseño Asistido por Computadora (<i>Computer Aided Design</i>).
<i>CA/D</i>	: <i>Converter Digital/ Analogue.</i>
<i>CAM</i>	: Manufactura Asistida por Computadora (<i>Computer Aided Manufacturing</i>).
<i>CCADET</i>	: Centro de Ciencias Aplicadas y Desarrollo Tecnológico.
<i>CCP</i>	: Captura-Comparación-PWM (<i>Capture-Compare-PWM</i>).
<i>C.I.</i>	: Circuito(s) Integrado(s).
<i>CN</i>	: Control Numérico.
<i>CNC</i>	: Control Numérico Asistido por Computadora.
<i>DDA</i>	: Analizador Diferencial Digital (<i>Digital Differential Analyzer</i>).
<i>DMOS</i>	: Transistores MOS de doble difusión (<i>doublé-diffused vertical MOS transistor</i>).
<i>DSM</i>	: Método por Búsqueda Directa (<i>Direct Search Method</i>).
<i>DSP</i>	: Procesador Digital de Señales (<i>Digital Signal Processor</i>).
<i>EDM</i>	: Maquinado por descarga eléctrica (<i>Electrical Discharge Machining</i>).
<i>EPROM</i>	: ROM programmable y borrable (<i>Erasable Programmable Read Only Memory</i>).
<i>EEPROM</i>	: ROM programmable y borrable eléctricamente (<i>Electrically Erasable Programmable Read-Only Memory</i>).
<i>FIB</i>	: Maquinado por haz de iones focalizados (<i>Focused Ion Beam</i>).
<i>FPGA</i>	: Arreglo de compuertas programables (<i>Field Programmable Gate Array</i>).

<i>GMM</i>	: Grupo de Micromecánica y Mecatrónica.
<i>gr</i>	: gramos.
<i>IBM</i>	: <i>International Business Machines</i> .
<i>ICD</i>	: Depurador en circuito (<i>In-Circuit Debugger</i>).
<i>LBM</i>	: Maquinado por laser (<i>Laser Beam Machining</i>).
<i>LED</i>	: Diodo Emisor de Luz (<i>Light-Emitting Diode</i>).
<i>MEMS</i>	: Sistemas Microelectromecánicos (<i>MicroElectroMechanical Systems</i>).
<i>MET</i>	: Tecnología de MicroEquipo (<i>MicroEquipment Technology</i>).
<i>mH</i>	: mili Henries.
<i>MH</i>	: Máquina(s) Herramienta.
<i>MHz</i>	: <i>Mega Herzios</i> .
<i>mm</i>	: milímetros.
<i>μm</i>	: micrómetros.
<i>MMH</i>	: Micromáquina(s) Herramienta.
<i>MMT</i>	: Tecnología de Micromáquina (<i>MicroMachine Technology</i>).
<i>MOS</i>	: Transistores Semiconductor de Metal Oxido (<i>Metal Oxide Semiconductor</i>).
<i>MSSP</i>	: Puerto Serial Maestro Sincrono (<i>Master Synchronous Serial Port</i>).
<i>MST</i>	: Tecnología de Microsistemas (<i>MicroSystems Technology</i>).
<i>OTP</i>	: Programable solo una vez (<i>Once-Time Programmable</i>).
<i>PC</i>	: Computadora Personal (<i>Personal Computer</i>).
<i>PIC</i>	: Controlador de Interfaz Periférico (<i>Peripheral Interface Controller</i>).
<i>PSP</i>	: Puerto Paralelo Esclavo (<i>Parallel Slave Port</i>).

PWM : Modulación por ancho de pulso (*Pulse-Width Modulation*).

ROM : Memoria de solo lectura (*Read-Only Memory*).

RPM : Revoluciones por minuto.

T.I. : *Texas Instruments*.

TQFP : *Thin Quad Flat Package*.

UNAM : Universidad Nacional Autónoma de México.

USART : Transmisor/Receptor Universal Síncrono/Asíncrono (*Universal Synchronous Asynchronous Receiver Transmitter*).

USD : Dólares norteamericanos.

V : Volts.

Referencias.

- [1] Leopoldo Ruiz Huerta. *Desarrollo de microequipo para células de manufactura*. División de estudios de posgrado, Facultad de Ingeniería, 2005. Tesis de doctorado. Área mecánica.
- [2] Kussul, E., Baydik, T., Ruiz, L., Caballero, A., Velazco, G. *Development of Low-cost Microequipment*. En 2002 IEEE International Symposium on Micromechatronics and Human Science. Nagoya Municipal Industrial Research Institute (Eds.), Nagoya, Japan.
- [3] Alberto Caballero Ruiz. *Metodología de evaluación para microequipo*. División de estudios de posgrado, Facultad de Ingeniería, 2005. Tesis de doctorado. Área mecánica.
- [4] Alberto Caballero Ruiz. *Caracterización de un microcentro de maquinado de bajo costo*. División de estudios de posgrado, Facultad de Ingeniería, 2000. Tesis de maestría. Ingeniería mecánica.
- [5] E. Kussul, D. Rachkovskij, T. Baydik and S. Talayev 1996. *Micromechanical engineering: a basis for the low cost manufacturing of mechanical microdevices using microequipment*. J. Micromech. Microeng.
- [6] German Herrera Granados. *Aplicación de sistemas CAD y CAM para el desarrollo de una micromáquina herramienta*. Facultad de Ingeniería. Ciudad Universitaria. Julio, 2007. Tesis de Licenciatura en Ingeniería mecánica.
- [7] Leopoldo Ruiz Huerta. *Diseño y construcción de un microcentro de maquinado de bajo costo*. División de estudios de posgrado, Facultad de Ingeniería, 2000. Tesis de maestría, Ingeniería mecánica.
- [8] Leopoldo Ruiz Huerta, Alberto Caballero Ruiz. Presentación del GMM, 2006-2007, CCADET, UNAM.
- [9] Okazaki, Kitahara. *Micro-machine Tool to machine micro-parts*, Proc. ASPE Annual meeting, 2000.
- [10] Angélica Zamora Vázquez. *Metodología para desarrollar motores a pasos*. Facultad de Ingeniería. Ciudad Universitaria. Setiembre, 2007. Tesis de Licenciatura en Ingeniería eléctrica-electrónica.
- [11] José Guillermo Saavedra Román. *Desarrollo de un sistema duplicador de pulso cardiaco para evaluación de bioprótesis cardiacas elaboradas en el Instituto Nacional de Cardiología*. Facultad de Ingeniería. Ciudad Universitaria. Setiembre, 2006. Tesis de Licenciatura en Ingeniería mecánica.
- [12] Alfredo Yáñez Soto. *Desarrollo de una interfaz computacional para el control de Micromáquinas-Herramienta*. Facultad de Ingeniería. Ciudad Universitaria, Marzo, 2007. Tesis de Licenciatura en Ingeniería en computación.
- [13] Ferré Masip R. *Como programar un control numérico*. Editorial Alfaomega. México, D.F. 1999. pags 155.

- [14] Y, Koren. *Interpolator for a Computer Numerical Control System*. IEEE Trans. On Computer, Vol. 25, No. 1, Jan. 1976, pp. 32 – 37.
- [15] A. Caballero Ruiz, H. Silva López, L. Ruiz Huerta. *Desarrollo de un sistema de control para una micromáquina herramienta basado en código estándar para CNC*. 8vo. Congreso Iberoamericano de Ingeniería Mecánica. Cusco, Perú. Octubre, 2007.
- [16] Y, Koren. *Design of Computer Control for Manufacturing Systems*. Faculty of Mechanical Engineering, Technion - Israel Institute of Technology, Haifa, Israel. Transactions of the ASME, Vol. 101, pag.326 August 1979.
- [17] Y, Koren. O, Masory. *Reference - Pulse Circular Interpolators for CNC Systems*. Faculty of Mechanical Engineering, Technion-Israel Institute of Technology, Haifa, Israel. Journal of Engineering for Industry, Vol. 103, pag. 131, February 1981.
- [18] Y, Koren., J, Ben-Uri. *Numerical Control of Machine Tools*. Kahanna Publishers, Delhi, India, 1978.
- [19] Y, Koren. O, Masory. *Reference - Word Circular Interpolators for CNC Systems*. Faculty of Mechanical Engineering, Technion-Israel Institute of Technology, Haifa, Israel. Transaction of the ASME, Vol.104, pag. 400, November 1982.
- [20] Y, Koren. *Computer Control of Manufacturing Systems*. Technion Israel Institute of Technology. Editorial McGraw-Hill, Inc. México, 2003.
- [21] J, E, Gorman. and J, Raamot. *Integer Arithmetic Technique for Digital Control Computers*. Computers Design, July 1970, pp. 51–57.
- [22] P. A. Sudarshan, *Electronic Techniques for NC Contour Machining*. 6th MTDR Conference, Dec. 1973.
- [23] A. S. Angelov, A. D. Mihailov, and G. N. Nachev. *Simple Algorithm for Curvilinear Interpolation*. 11th Annual Meeting and Technical Conference Numerical Control Society, Proceedings, March 31 April 3, 1974, pp. 338-348.
- [24] Carlos Muñoz Leines., César Santos Carrasco. *Empleo de FPGA's para el sistema de control de motores a pasos bimodales en micromecánica*. Facultad de Ingeniería. Ciudad Universitaria. Mayo, 2003. Tesis de Licenciatura en Ingeniería Eléctrica Electrónica.
- [25] Héctor Hugo Silva López. *Automatización de una micromáquina herramienta de primera generación*. Facultad de Ingeniería. Ciudad Universitaria, Septiembre, 2005. Tesis de Licenciatura en Ingeniería Eléctrica-Electrónica.
- [26] Oscar Cuevas Gutiérrez. *Diseño de un sistema de control de una microfábrica basado en un DSP*. Facultad de Ingeniería. Ciudad Universitaria. Septiembre, 2007. Tesis de Licenciatura en Ingeniería Eléctrica-Electrónica.
- [27] Smid P. *A Comprehensive Guide to Practical CNC Programming. CNC PROGRAMMING HANDBOOK*. Editorial Industrial Press, Inc. Segunda edición. Nueva York, EUA, 2003.
- [28] Oscar Pérez Reyes. *Desarrollo de motores de pasos para micromáquinas herramienta*. Facultad de Ingeniería. Ciudad Universitaria, Facultad de Ingeniería. Junio, 2008. Tesis de Licenciatura en Ingeniería Eléctrica-Electrónica.
- [29] José Reséndiz Sánchez. *Investigación del sistema de control de una MMH y propuesta de optimización del sistema*. Facultad de Ingeniería. Ciudad Universitaria, Marzo, 2005. Tesis de Licenciatura en Ingeniería mecánica electricista.

- [30] Iván Escalona. *Diseño y manufactura asistidos por computadora. Introducción al CNC*. Ingeniería industrial, UPIICSA-IPN, México D.F. www.monografias.com
- [31] <http://www.faulhaber-group.com/n380604/n.html>
- [32] <http://www.sandia.gov/media/NewsRel/NR2001/minirobot.htm>
- [33] <http://www.linuxelectrons.com/news/general/epson-announces-advanced-model-worlds-lightest-micro-flying-robot>
- [34] E. M. Kussul, L. M. Kasatkina. *Los problemas de desarrollo de máquinas herramienta controladas con computadora para fábricas micro mecánicas sobre mesa*. // Sistemas de control y computadoras. 1998, N5, V.28, pp 32-39.
- [35] M, Tanaka. *Development of desktop machining microfactory*. Riken Review. No. 34, Abril, 2001. Japón.
- [36] Alberto Caballero R. Leopoldo Ruiz H. *Apuntes de diseño y construcción de prototipos*. Programa de estudios de maestría, periodo escolar 2005-2. Ciudad Universitaria.
- [37] Ricardo Jiménez. *Control Numérico por Computadora (CNC)*. Ingeniería de manufactura.
- [38] http://www.mech.mcmaster.ca/undergraduate/courses/me3m02/pdf_files/me302_expt_p3.pdf. Experiment 3 CNC Machining.
- [39] Atila Ertas, Jesse C. Jones. *The Engineering Design Process*. Department of Mechanical Engineering Texas Tech University. Editorial Jhon Wiley & Sons, Inc. 1993.