



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

**PROGRAMA DE MAESTRÍA Y DOCTORADO
EN INGENIERÍA**

FACULTAD DE INGENIERÍA

**DESARROLLO DE UN SISTEMA DE CONTROL
NUMÉRICO POR COMPUTADORA PARA UNA
MICROMÁQUINA HERRAMIENTA.**

TESIS

QUE PARA OBTENER EL GRADO DE:

MAESTRO EN INGENIERÍA

MECÁNICA-MECATRÓNICA

PRESENTA:

ING. SIMÓN MARTÍNEZ CRESCENCIO

TUTOR:

DR. ALBERTO CABALLERO RUIZ

CIUDAD UNIVERSITARIA , FEBRERO 2013.



JURADO ASIGNADO:

Presidente : Dr. Marcelo López Parra
Secretario : Dr. Leopoldo Adrián González González
Vocal : Dr. Alberto Caballero Ruiz
1^{er} Suplente : Dr. Leopoldo Ruiz Huerta
2^o Suplente : Dr. José Sánchez Vizcaíno

Lugar donde se realizó la tesis:

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

DIRECTOR DE TESIS:

DR. ALBERTO CABALLERO RUIZ

Firma

AGRADECIMIENTOS

Este trabajo se desarrolló en el Centro de Ciencias Aplicadas y Desarrollo Tecnológico de la Universidad Nacional Autónoma de México, con la tutoría del Dr. Alberto Caballero Ruiz del Grupo de Micromecánica y Mecatrónica.

AGRADECIMIENTOS

A Dios, por permitirme alcanzar la luz que parecía no encontrarla.

A mis Padres, por brindarme su apoyo incondicional y enorme sacrificio de toda la vida, por darme un gran ejemplo de lucha y perseverancia, gracias por su comprensión en los buenos y malos momentos.

A mis hermanos, por sus consejos y motivación de seguir adelante.

A mi camarada y amigo Tomás Santos, por sus consejos, enseñanzas y apoyo fundamental que ha sido determinante para el logro de mi objetivo y que sin el cual no hubiese sido posible, no tengo palabras para agradecerte, gracias amigo.

A mi tutor Alberto Caballero, gracias por su confianza, consejos, enseñanzas, por haberme tenido bastante paciencia y dejarme formar parte de este grupo de trabajo, gracias por todo lo que aprendí.

A Alberto Ferrusquía gracias por sus consejos, conocimiento y por contribuir en el desenlace de este trabajo.

A mis compañeros y amigos del GMM, Omar León, Alejandro, Emilio, Marcos, Iván, Hugo y Eduardo con los cuales compartí el laboratorio y por hacer de este lugar un ambiente enriquecedor y de aprendizaje.

No menos importante, sí no fundamental es para mí agradecer al CONACYT que brindó el apoyo económico para mis estudios.

A mis sinodales, gracias por las correcciones, comentarios y aportaciones a este trabajo y a mi formación profesional.

A la UNAM, gracias por abrirme sus puertas y darme la oportunidad de realizar un posgrado.

CONTENIDO

INTRODUCCIÓN.....	1
1. ANTECEDENTES.....	3
1.1 Sistemas de Control para MMH desarrollados en el GMM.....	5
2. DESCRIPCION DEL PROBLEMA, OBJETIVO Y ALCANCES.....	11
2.1 Descripción del problema.....	14
2.2Objetivo.....	15
2.3Alcances.....	15
3. MARCO TEÓRICO.....	16
3.1 Fundamentos de Control Numérico por Computadora.....	16
3.1.2 Definición de CNC.....	16
3.1.3 Elementos de una Máquina CNC.....	17
3.2 Técnicas de Control en Sistemas CNC.....	21
3.2.1 Técnica por referencia de pulsos.....	21
3.2.2 Técnica por datos muestreados.....	22
3.3 Métodos de Interpolación por referencia de pulsos.....	24
3.3.1 Interpolación lineal por DDA.....	24
3.3.2 Interpolación circular por DDA.....	27
3.3.3 Interpolación circular incremental.....	28
3.3.4 Interpolación circular por búsqueda directa.....	32
4. DESARROLLO DEL SISTEMA DE CONTROL.....	35
4.1 Características requeridas del sistema de control.....	35
4.2 Propuesta de control.....	36
4.3 Selección de Hardware electrónico.....	36

4.3.1 Driver A3987.....	39
4.3.2 Driver R-325.....	41
4.3.3 Evaluación de los Drivers de potencia.....	43
4.3.4 Selección del modelo de Microcontrolador.....	44
4.4 Características del Sistema de Control.....	46
4.4.1 Modos de Operación de la MMH.....	46
4.4.2 Parámetros monitoreados.....	47
4.4.3 Códigos G y M reconocidos.....	48
4.4.4 Sintaxis de Programación.....	48
4.5 Selección de los algoritmos de Interpolación.....	49
4.5.1 Simulación de los algoritmos de interpolación.....	49
4.5.2 Evaluación de los algoritmos de interpolación.....	57
4.6 Desarrollo de la interfaz gráfica.....	58
4.7 Estructura del programa de control.....	62
4.8 Ejemplos de operación del programa de control.....	63
4.8.1 Ejemplo (1).....	64
4.8.2 Ejemplo (2).....	66
5. PRUEBAS Y RESULTADOS.....	68
5.1 Pruebas de Frecuencia de ejecución de los Algoritmos de interpolación.....	69
5.1.1 Ejecución de los Algoritmos Usando Motores de la MMH Tipo Puente.....	69
5.1.2 Ejecución de los Algoritmos Usando la 2da Versión del Primer Prototipo de MMH.....	73
5.2 Maquinado con MMH tipo Puente para Determinar Juego Mecánico.....	74
5.3 Maquinados con 2da Versión del Primer Prototipo de MMH.....	76
5.4 Maquinados con Interpolación en dos Ejes Simultáneos.....	87
5.5 Procedimiento Para la Ejecución de un Programa CNC.....	90
5.5.1 Diseño en CAD.....	91

5.5.2 Proceso de Manufactura.....	91
5.5.3 Ejecución de programas CNC.....	93
CONCLUSIONES.....	98
TRABAJO A FUTURO.....	100
REFERENCIAS.....	101

INTRODUCCIÓN

En la actualidad la automatización de sistemas mecánicos contribuyen en gran medida a la ejecución de tareas de una manera mucho más rápidas y con mayor precisión que cualquier ser humano. La automatización de estos sistemas ha tenido un efecto enorme en el mundo en que vivimos,desempeñando un papel importante en la productividad y la reducción de costos en la industria manufacturera. Un tipo de automatización es el Control Numérico por Computadora(CNC) de las Máquinas Herramienta (MH), que ejecutan operaciones para la fabricación de partes o productos.

A nivel mundial, la creciente tendencia hacia la miniaturización ha impactado tecnológicamente enprácticamente casi todos los campos de la industria, desde la medicina hasta la manufactura. Con el incremento de productos más compactos, la demanda en la precisión de componentes miniaturizados es alta y la capacidad para la manufactura de estos componentes con exactitud es crítica para los fabricantes que desean mantenerse competitivos en el mercado [1].

El enfoque de la Ingeniería Micromecánica es la fabricación de dispositivos micromecánicos automatizados con dimensiones inferiores al milímetro. También propone mediante el uso de las técnicas convencionales de manufactura, la construcción de dispositivos empleando máquinas comparables en tamaño con las cuales estos dispositivos son producidos y que tengan aplicación en diversos campos de la industria, medicina, biotecnología, industria química, aeroespacial, etc. [2].LaMicromecánica como su nombre lo indica es el equivalente a la mecánica en general pero a una escala mucho más pequeña. Más allá de la mecánica convencional de precisión que opera con piezas de milímetros o décimas de milímetro, la Micromecánica llega hasta tamaños de micrómetros.

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. Esta investigación comenzó con la idea de que para fabricar micropartes mecánicas deben emplearse máquinas miniatura [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 producciones totalmente automatizadas, enfocadas a la creación de equipo e instrumentación de bajo costo y alta eficiencia [4,15].

El fundamento propuesto por el GMM consiste en crear generaciones de microequipo en las cuales los componentes de cada generación son producidos por la generación 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 manufacturada con ayuda del microequipo de la generación que lo precede [5,15].

Una tarea fundamental para lograr los objetivos mencionados es la automatización de los procesos de micromanufactura. Mediante la automatización de estos procesos se alcanzará eficiencia y precisión en los microequipos a desarrollar, tanto para su manufactura como su operación [6,7].

En este trabajo se propone el desarrollo de un sistema de Control Numérico por Computadora (CNC) para una micromáquina herramienta capaz de interpretar código estándar G y M, para su aplicación en procesos de micromanufactura.

El diseño de un sistema CNC requiere primeramente de la selección apropiada de la técnica de control para generar las tareas de interpolación lineal y circular las cuales controlan los movimientos de cada uno de los ejes de la máquina, desplazando la pieza o herramienta en trayectorias específicas.

El contenido de este trabajo se estructura en cinco capítulos principales.

El *capítulo 1* cubre la parte de los antecedentes generales sobre el desarrollo de sistemas micromecánicos en el mundo y en México, en seguida se incluyen las diferentes propuestas de control implementadas en el GMM anteriores a este trabajo. En el *capítulo 2*, se plantea el problema a resolver, el objetivo y los alcances de este trabajo. En la parte del *capítulo 3*, se efectúa una descripción de los fundamentos de sistemas CNC y de las técnicas de control comúnmente utilizadas. En el *capítulo 4* se describe el desarrollo de este trabajo, así mismo se muestra el análisis de los diferentes algoritmos de interpolación para determinar cuál es la mejor opción para ser aplicados en el desarrollo de un sistema CNC. Se da una descripción de las características de la micromáquina a controlar y se explica el desarrollo de una interfaz computacional para la comunicación entre el usuario y máquina.

Por último, en el *capítulo 5* se reportan las pruebas realizadas tanto al hardware electrónico y micromáquina. Finalmente, se incluye una sección de conclusiones de los resultados obtenidos y en otro apartado se plasma el trabajo a futuro donde se proponen mejoras al sistema.



ANTECEDENTES

En muchas áreas de la industria existe una fuerte tendencia hacia la miniaturización de productos. Predominan diversas tecnologías con diferentes nombres para la manufactura de los componentes micromecánicos de estos productos, por ejemplo los Sistemas Electromecánicos (MEMS-Electro Mechanical Systems) desarrollados principalmente en los Estados Unidos de América, Tecnología de Microsistemas (MST- Micro Systems Technology) cuyo mayor auge se encuentra en los países europeos y las Tecnologías de MicroMáquinas (MMT-Micro MachineTools) en Japón [8].

La tecnología empleada en microelectrónica, como la utilizada en los MEMS, permite crear dispositivos de bajas dimensiones con componentes mecánicos simples, sin embargo estos dispositivos poseen prácticamente dos dimensiones debido a su espesor [10].

Sin embargo, el desarrollo de sistemas micromecánicos más elaborados, como las micromáquinas herramienta, los micromanipuladores y los microrobots requieren de estructuras mecánicas con tres dimensiones reales, formas complejas, posibilidad de ensamble y movimiento, así como el uso de diferentes materiales [7].

Esta situación claramente impone la necesidad del desarrollo de nuevas tecnologías que pueda combinar las ventajas de los métodos de manufactura tradicionales existentes.

Enfocado en el uso de tecnologías mecánicas convencionales para generar micromecanismos, cuando deseamos obtener las tolerancias requeridas, se tienen dos alternativas a seguir: una es empleando *máquinas superprecisas* [11], que desafortunadamente, para aplicación comercial no es del todo recomendable

debido a que su costo de inversión es sumamente elevado; y la otra es *disminuyendo el tamaño de las MH* y manipuladores involucrados en su manufactura, la disminución en el costo de fabricación de los microdispositivos selogrará con base en la producción en masa y operación en paralelo de estasmicromáquinas [10]. Esto último puede observarse en el diagrama de la figura 1.1.

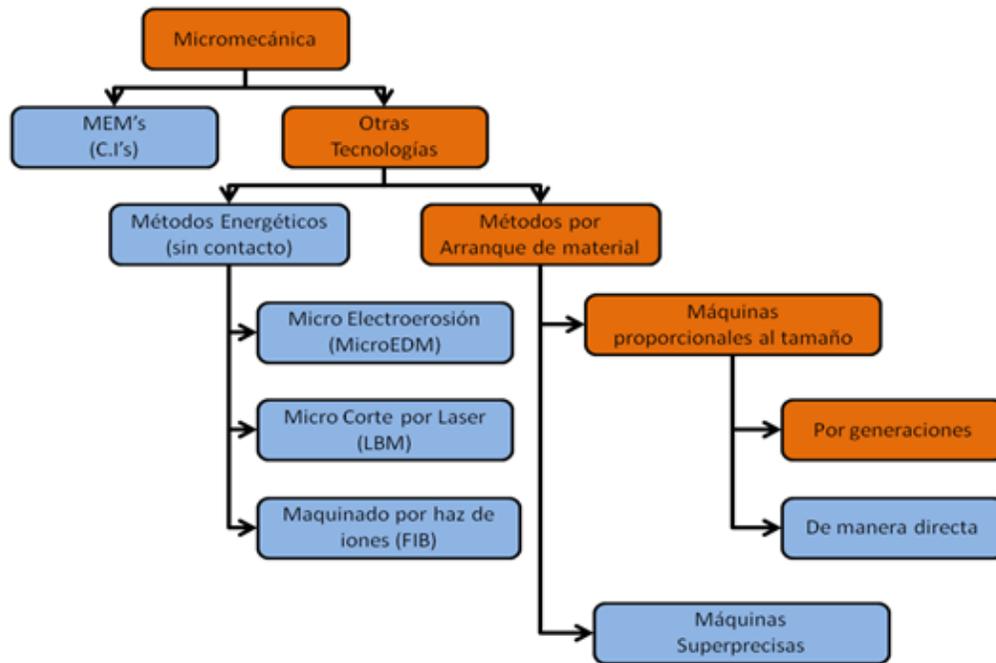


Figura 1.1. Desarrollo de la Micromecánica en el GMM [12].

En la figura 1.1, se puede apreciar en color naranja la ruta que el GMM ha seguido a lo largo de los años para el desarrollo de microequipo. Como se mencionó anteriormente, el plan global propuesto por el GMM, para generar micromáquinas a bajo costo, es desarrollar generaciones de microequipo en los cuales los componentes de cada generación son producidos por la generación anterior. Esta tecnología es denominada Tecnología de Microequipo (MET-MicroEquipment Technology) [2, 5].

1.1 Sistemas de Control para MMH desarrollados en el GMM

Desde que se construyó el primer prototipo de MMH en el año 2000, se han implementado diferentes sistemas de control. El sistema de control utilizado para el primer prototipo de MMH, fue diseñado con el objetivo de usar una computadora personal (PC) y a través de dos puertos paralelos controlar la posición y velocidad de los motores que accionan los ejes de la máquina, evitando en la medida de lo posible hardware de control externo. El sistema consta de un amplificador de DC de 16 canales, y cuatro acondicionadores de señal para los sensores de contacto colocados al inicio de carrera de cada eje [21].

El desarrollo de una interfaz gráfica fue elaborado con el software Borland C++, mostrada en la figura 1.2, mediante esta interfaz se monitorea el estado de la MMH a través de dos puertos paralelos que se encuentran divididos en tres secciones, dos de ellas pertenecen a las conexiones que comunican a los actuadores y la otra a las señales de retroalimentación provenientes de los sensores de posición de origen por contacto físico y por contacto eléctrico [21].

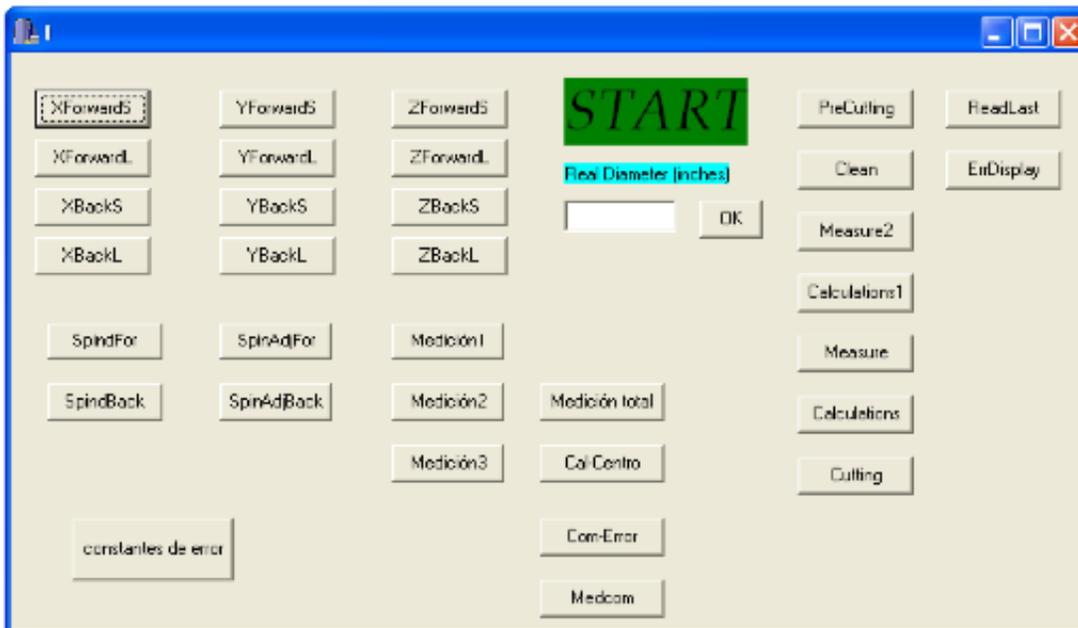


Figura 1.2. Interfaz de Control empleado en el primer prototipo [21].

Con este sistema de control se fabricaron diversas piezas de manera automática, las cuales podemos observar en la figura 1.3.

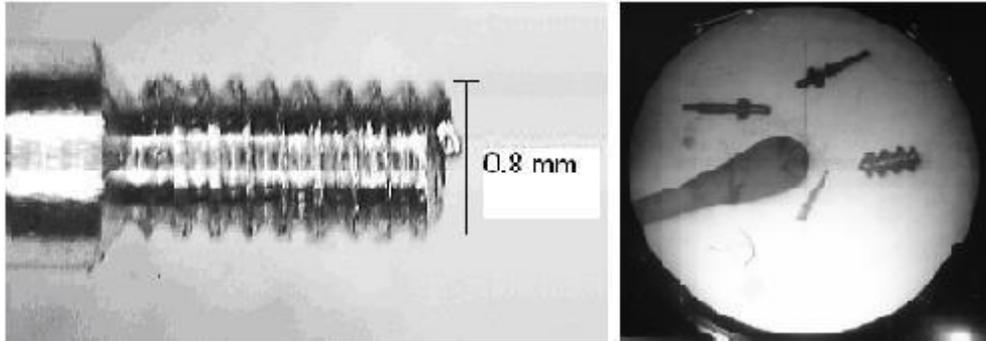


Figura 1.3 Micropiezas fabricadas con el primer prototipo [40].

Un segundo sistema de control implementado, consiste de una interfaz computacional desarrollada en Visual Basic [22], la cual es capaz de interpretar código G y M estándar para CNC. Este sistema de control utiliza los mismos elementos de hardware que el primer sistema de control, mostrados en la figura 1.4.

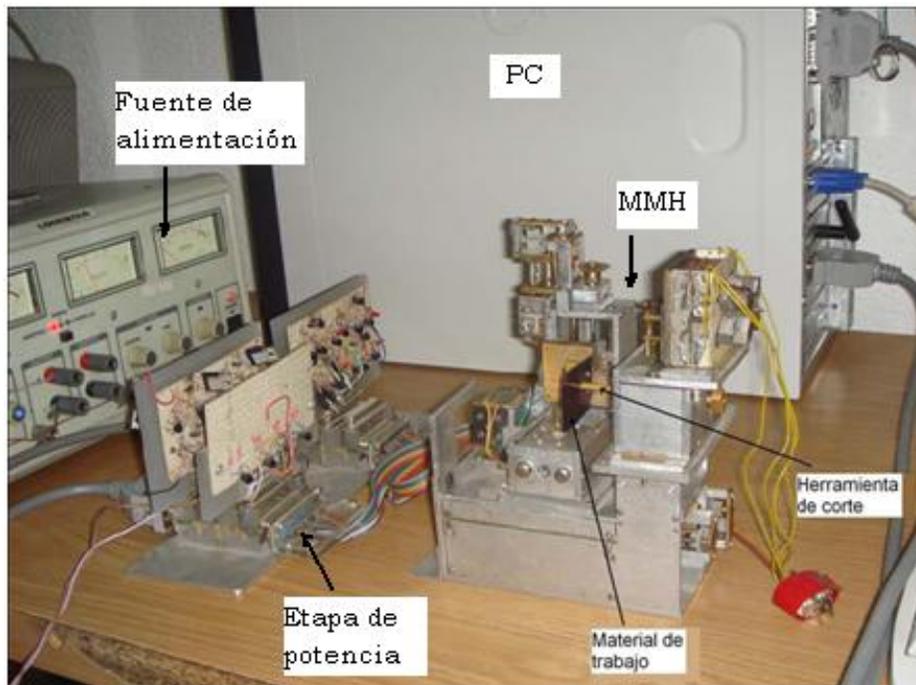


Figura 1.4 Componentes empleados para el control por medio de la interfaz computacional [24]

La interfaz se muestra en la figura 1.5, se encarga de recibir y mostrar la información proveniente de la PC, consta de cinco secciones: un *menú* para abrir

un archivo que contenga el programa de la pieza, tipo de conexión del sistema y configuración, un *cuadro de imagen* que ilustra el plano XY para mostrar gráficamente las trayectorias de la herramienta, un *cuadro de lista* para escribir las instrucciones del programa de la pieza, un *área de despliegue de coordenadas* relativas al cero máquina y una *sección de control* que contiene los modos de operación de la MMH [22].

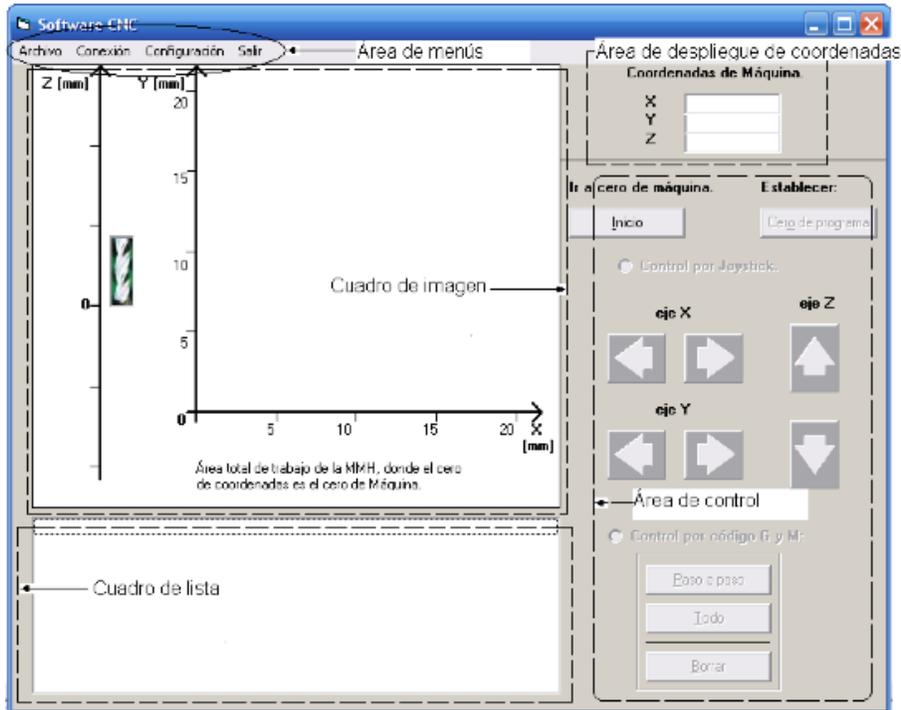


Figura 1.5. Pantalla principal de la interfaz computacional para el control de una MMH [22].

El procedimiento para maquinarse una pieza utilizando esta aplicación es en primera instancia generar mediante un software CAD el dibujo de la pieza final y con ayuda de un software CAM importar el dibujo y generar el código correspondiente. Posteriormente del menú de la interfaz, abrir el archivo del código generado.

Con esta aplicación se realizaron maquinados de figuras en donde se emplean interpolaciones en dos ejes. Se ejecutaron dos programas con código G y M, que representan el escudo de los Pumas con dimensiones de 4mm y el logo del GMM dimensionado en 12mm. Los resultados se muestran en la figura 1.6.

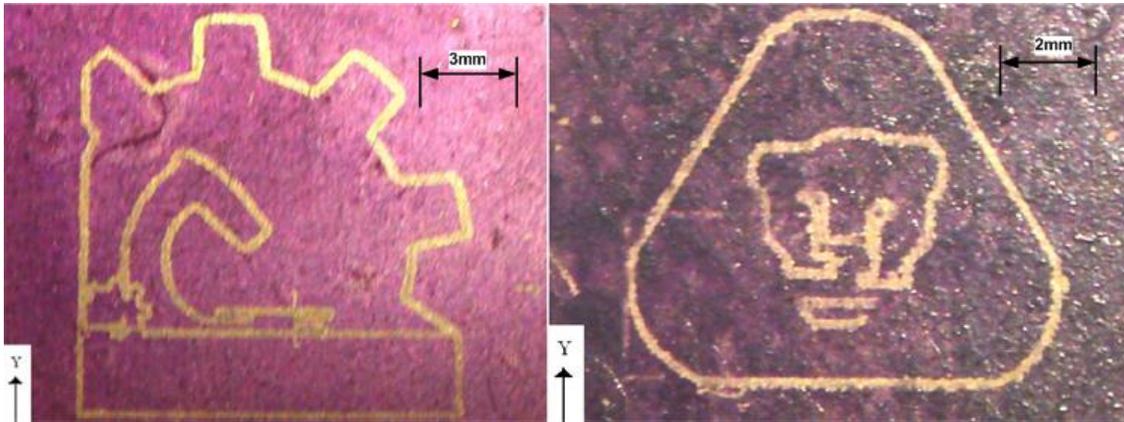


Figura 1.6. Maquinados producidos por la aplicación computacional realizada en Visual Basic [22].

El tercer sistema de control se implementó mediante el uso de una PC y las características con que cuenta este sistema son:

- a) Reconoce instrucciones G y M para procesos de fresado.
- b) Cuenta con cuatro modos de operación para accionar el movimiento en la MMH:
 - a) Por Joystick.
 - b) Por Secuencia de Pasos.
 - c) Por ejecución de una línea de código G y M.
 - d) Por ejecución de un programa de código G y M.
- c) Los algoritmos de control están programados en un microcontrolador PIC.
- d) Cuenta con una interfaz gráfica para el monitoreo de los diferentes modos de operación así como también de las trayectorias y parámetros del proceso de manufactura (avance, rpm, posición).

La comunicación usuario-máquina se realiza a través de una interfaz realizada en Visual Basic, la cual se muestra en la figura 1.7.

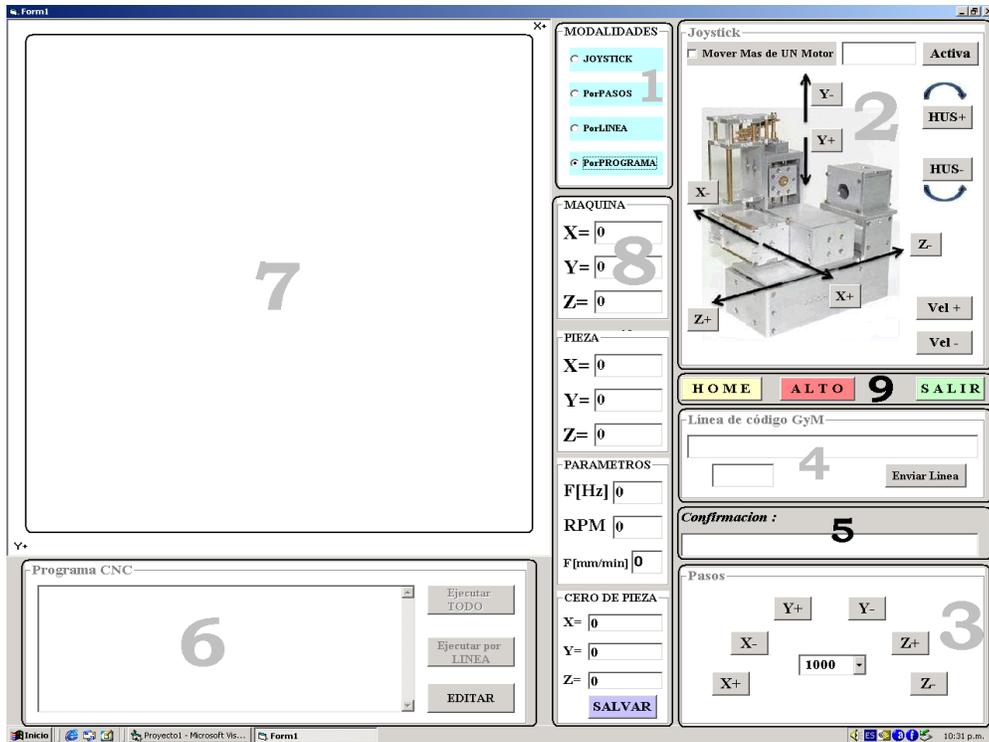


Figura 1.7. Interfaz Computacional para la 2ª versión del primer prototipo de MMH [24].

En el diagrama de la figura 1.8 se muestra el sistema de control desarrollado para una MMH de primera generación, mediante algoritmos de control programados en un dispositivo externo a la PC en este caso un microcontrolador PIC.

La tarjeta controladora cuenta 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 (etapa de potencia) proporcione los niveles de corriente adecuados a las señales de referencia generadas (control) para el accionamiento de los actuadores [24].

Como podemos observar en la figura 1.8 el sistema de control es de lazo abierto, en este caso, cada pulso generado produce un desplazamiento equivalente a la resolución de la MMH, el número acumulado de pulsos representa la posición y la frecuencia de los pulsos es proporcional a la velocidad de cada eje[25].

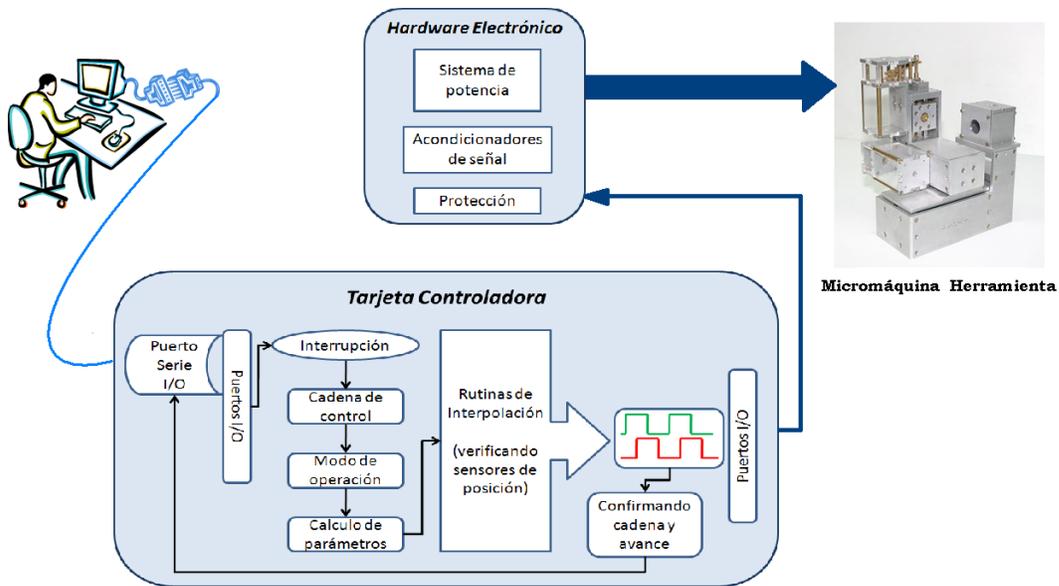


Figura 1.8. Sistema de Control para una MMH [24].

Con este sistema de control se realizaron maquinados de contorno del logo de los pumas cuya dimensión es de 5mm y el logo del GMM con dimensiones de 12 mm. En la figura 1.9 podemos observar dichos maquinados.

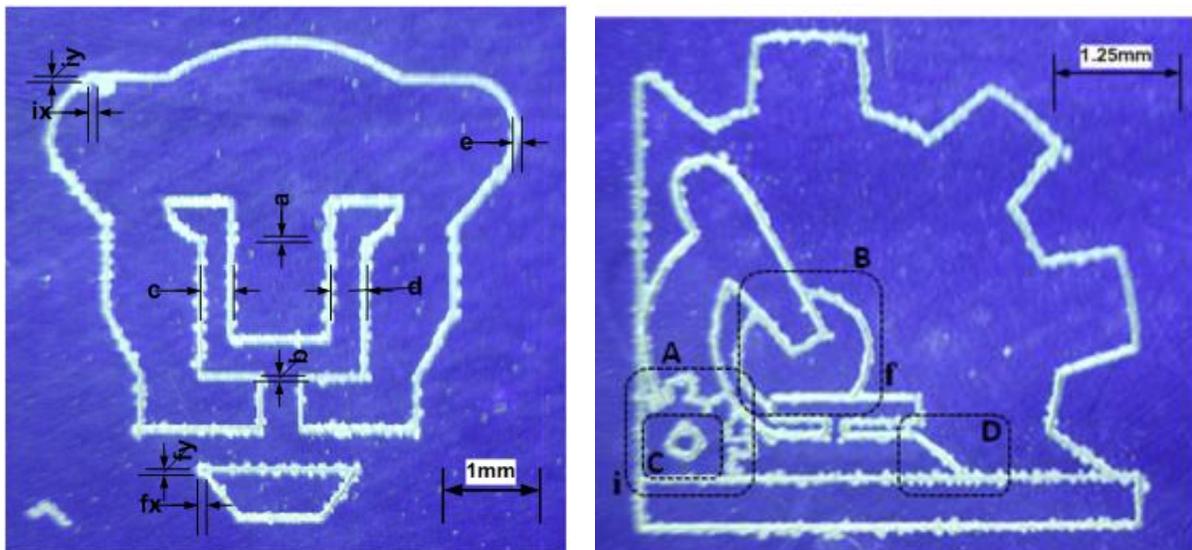


Figura 1.9. Maquinados del logo de los pumas y del GMM [24].



DESCRIPCIÓN DEL PROBLEMA, OBJETIVO Y ALCANCES

La Máquina Herramienta (MH) es uno de los elementos que ha jugado un papel fundamental en el desarrollo tecnológico en el mundo y podemos decir sin exagerar que ha afectado directamente en el desarrollo tecnológico industrial. Gracias a la utilización de la Máquina Herramienta se ha podido realizar maquinaria de todo tipo.

Hasta la década de los 50's, la mayor parte de las operaciones de manufactura era efectuada con maquinaria tradicional como tornos, fresadoras y prensas, que carecían de flexibilidad y necesitaban de mucha mano de obra capacitada [26].

Debido a que día a día aparecieron nuevas necesidades y requerimientos, forzó la utilización de nuevas técnicas que reemplazaran al operador humano. De esta forma se introdujo el Control Numérico en los procesos de fabricación, impuesto por varias razones:

- Necesidad de fabricar productos que no se podían conseguir en cantidad y calidad suficientes.
- Necesidad de obtener productos hasta entonces muy difíciles de fabricar por ser excesivamente complejos.
- Necesidad de bajar los costos de producción para ser competitivos.

Estas circunstancias obligaron a mejorar la eficiencia y la flexibilidad de las operaciones de manufactura.

Con el uso de sistemas CNC en MH se pueden realizar operaciones de manufactura con facilidad para producir piezas en forma repetitiva y exacta, así como también manejar piezas distintas (con solo cargarles programas para partes distintas). El proceso de la información, cálculo de velocidades y las interpolaciones entre dos puntos son realizados por el programa de control [26].

En la actualidad los investigadores requieren de técnicas de fabricación que puedan producir económica y eficientemente componentes de tamaño micrométrico con formas complejas de tres dimensiones (3D) en un amplio rango de tipos de materiales. Comparado con los procesos de maquinado en escala macro, con el micromaquinado tal como el microfresado es posible crear complicadas estructuras de varios materiales utilizando microherramientas para la remoción de material [27].

En el mundo, varios investigadores y compañías se han dado a la tarea de escalar máquinas herramienta para producir microcomponentes. Entre varios grupos de investigación destaca el Laboratorio de Ingeniería Mecánica (MEL- Mechanical Engineering Laboratory) del Instituto Nacional de Ciencia y Tecnología Avanzada (AIST – National Institute of Advanced Industrial Science and Technology) de Japón, que comenzó el desarrollo de esta tecnología con la idea de que *Máquinas miniatura debe emplearse para fabricar micropartes mecánicas*. La integración de estos sistemas condujo al concepto de microfábrica sobremesa, que se llevó a la realidad en la década de los 90' s [15].

En la figura 2.1, se muestra una microfábrica sobremesa automatizada desarrollada en el año 2000 por investigadores del Laboratorio de Ingeniería Mecánica (MEL) de la AIST, en Japón. Un ejemplo de un producto fabricado por esta microfábrica es un rodamiento de bolas miniatura de 0.9 mm de diámetro exterior y con flecha de 100 μm de diámetro [3].

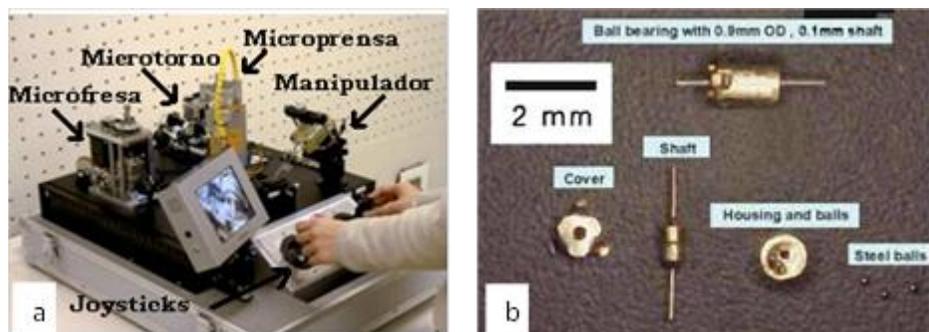


Figura 2.1. a) Microfábrica automatizada. b) Rodamiento de bolas miniatura.

En el Laboratorio de Instrumentación y Tecnología de Medición de Precisión del Centro de Micronanomanufactura y Tecnología de la Universidad de Tiajin en China [44], desarrolló una microfresadora de 3 ejes sobremesa. Para determinar el desempeño de micromaquinado de esta micromáquina se maquinaron dos engranes en material de cobre con diámetros 5.4 mm y 1.35 mm con espesor de 1 mm, ver figura 2.2.

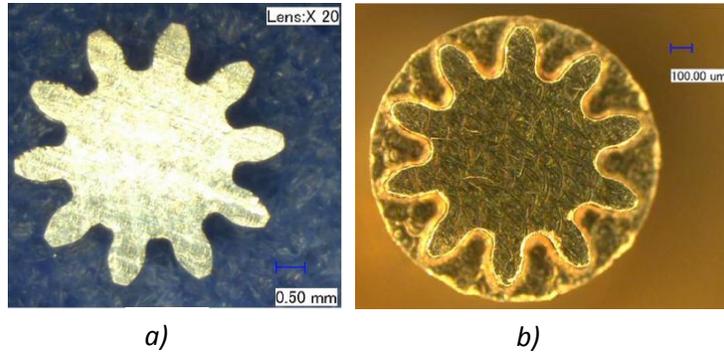


Figura 2.2. Perfil de 2 engranes: (a) engrane de 5.4 mm y (b) engrane de 1.35 mm

Microlution Inc. Es un fabricante especializado en la construcción de micromáquinas herramienta CNC de alto desempeño para la manufactura de micropiezas de alta precisión en una amplia variedad de materiales. La exactitud de estas micromáquinas están en el orden de $\pm 1 \mu m$. Ejemplos de partes que se pueden fabricar con estas micromáquinas son: barrenos de 0.203 mm de diámetro fabricados en placa de latón ver figura 2.3, boquillas con diámetro de $70 \mu m \pm 1 \mu m$ para clasificación de células, ver figura 2.4.

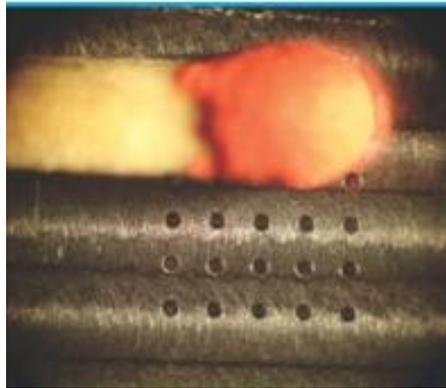


Figura 2.3. Barrenos de 0.203 mm de diámetro.



Figura 2.4. Boquilla para clasificación de células.

Otro fabricante de micromáquinas herramienta es la compañía Datron Dynamics, los centros de maquinado de esta marca de micromáquinas pueden producir implantes de plástico interoculares para operación en ojos, estos son manufacturados con tolerancias de $\pm 5 \mu m$. La compañía argumenta que sus máquinas proveen una precisión en la posición de $\pm 3 \mu m$ y una repetibilidad de $\pm 2 \mu m$.

El micromaquinado es un método para la creación de dispositivos y componentes miniatura de dimensiones que van desde decimas de micrómetros hasta unos pocos milímetros [29].

El sistema de control para una micromáquina CNC es especialmente importante para asegurar su funcionamiento. En efecto para lograr un buen desempeño de maquinado y proveer una buena precisión en la posición, es necesario desarrollar un apropiado sistema CNC [28].

2.1 Descripción del Problema

El GMM ha trabajado en el desarrollo de nuevas Micromáquinas Herramienta (MMH) para procesos de microfresado por lo que se ha enfocado en la implementación de sus propios sistemas CNC en sus prototipos de MMH, con la finalidad de simplificar y facilitar sus métodos de programación en la manufactura de micropiezas.

En el trabajo *Desarrollo de un Sistema de Control Para Una Micromáquina Herramienta Basado en Código Estándar para CNC*[24], aunque cumple con el objetivo de interpretar código estándar G y M, presenta algunos inconvenientes, como por ejemplo la frecuencia máxima de operación de los actuadores de la micromáquina es de 170 Hz, con esto, no es posible utilizar la técnica de micropasos para diferentes resoluciones de los motores, también representa una baja velocidad durante el proceso de maquinado y por consiguiente el tiempo en la manufactura de piezas se incrementa.

En lo que respecta a la interfaz gráfica, su versatilidad presenta algunas deficiencias, como por ejemplo no es posible importar o abrir un programa CNC desde cualquier localidad dentro de la PC. Otra desventaja presente es en la programación, para definir el tipo de operación que ha de ejecutar la máquina, es necesario programar un código G en cada bloque, por lo que no sigue la regla de los códigos modales, es decir un código debe permanecer activo hasta que otro del mismo grupo lo reemplace, por lo tanto esto nos lleva a una programación ineficiente.

Por tal motivo en este trabajo se plantea una propuesta para tratar de solucionar estos problemas, así como también lograr la flexibilidad de utilizar el control en los diferentes prototipos de MMH que usan como actuadores motores de pasos con diferentes resoluciones.

2.2 Objetivo

Desarrollar un sistema de control que sea capaz de interpretar instrucciones de código G y M estándar para su aplicación a una Micromáquina Herramienta para la ejecución de procesos de micromanufactura.

2.3 Alcances

- Investigar las técnicas de control y algoritmos de interpolación empleados en sistemas de control numérico por computadora.
- Comparar y seleccionar, con base en simulaciones, los algoritmos de interpolación a emplear.
- Comparar y seleccionar dispositivos electrónicos para la integración de una tarjeta de control y etapa de potencia del sistema de control.
- Implementar la etapa de control y de potencia.
- Desarrollar una Interfaz gráfica para simulación y comunicación usuario-máquina, con la flexibilidad de poder seleccionar las características de los motores de pasos de la micromáquina.
- Hacer pruebas de comunicación entre la PC, tarjeta de control y motores de pasos de la MMH mediante la implementación de los algoritmos de interpolación elegidos.
- Realizar pruebas de maquinados efectuando movimientos en un solo eje a la vez, para determinar las características de posicionamiento de la micromáquina a controlar.
- Llevar a cabo las mediciones de los maquinados obtenidos para compensar el juego mecánico existente en cada uno de los ejes.
- Realizar pruebas de maquinados de contornos de figuras donde esté presente el movimiento simultáneo de dos ejes.
- Analizar los resultados, plasmar las conclusiones y plantear trabajo a futuro.

En el capítulo 3, se establece el marco teórico referente a los principios de operación, técnicas de control y algoritmos de interpolación empleados en los sistemas CNC.



MARCO TEÓRICO

La evolución de la tecnología de las máquinas herramienta ha estado marcada por grandes avances en la capacidad de control, particularmente en los últimos 30 años, con la llegada del control numérico computacional y avances relacionados han traído importantes cambios y efectos en los métodos de manufactura y sus costos. Un equipo CNC se caracteriza por contar con una computadora integrada a la máquina, en donde se pueden conservar una gran cantidad de programas de trabajo en forma digital por disponer de memoria de almacenamiento, también de facilitar su edición, modificación o corrección en pocos minutos, además con la gran ventaja de diagnosticar las órdenes antes de ejecutarlas, y en caso de un error nos previene antes de llevarlas a cabo.

El CNC es una forma especializada y versátil de automatización por software y su aplicación cubre muchos tipos, sin embargo inicialmente fue desarrollado para control de movimientos y operaciones de máquinas herramienta [30].

3.1 Fundamentos de Control Numérico por Computadora

3.1.1 Definición de CNC

Según la Asociación Industrial Electrónica (EIA –Electronic Industry Association) el término CNC lo define como un sistema en el cual las acciones son controladas por el ingreso directo de datos numéricos. El sistema debe automáticamente interpretar al menos alguna parte de estos datos. En pocas palabras un sistema CNC recibe datos numéricos, los interpreta y posteriormente realiza el control de las acciones establecidas.

Y de acuerdo con [27] también define CNC como un sistema de automatización programable que se basa en una serie de instrucciones codificadas (letras y números) que gobiernan todas las acciones de una máquina haciendo que ésta desarrolle una secuencia de operaciones y movimientos previamente establecidos.

En la figura 3.1, se muestran los componentes básicos de un sistema CNC, los cuales son: *un programa*, que consiste en una secuencia de instrucciones 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 [27].

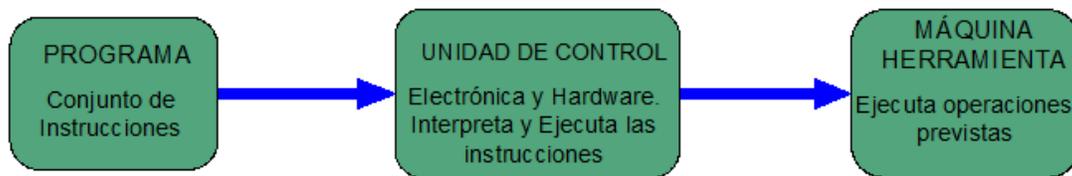


Figura 3.1 Elementos básicos de un sistema CNC.

3.1.2 Elementos de una Máquina CNC

En el diagrama de la figura 3.2 se muestran los elementos de una máquina CNC que consiste de seis componentes principales:

- a) Dispositivo de entrada.
- b) Unidad de Control de la Máquina.
- c) Máquina Herramienta.
- d) Sistema de Actuadores.
- e) Dispositivos de Retroalimentación.
- f) Unidad de Visualización.

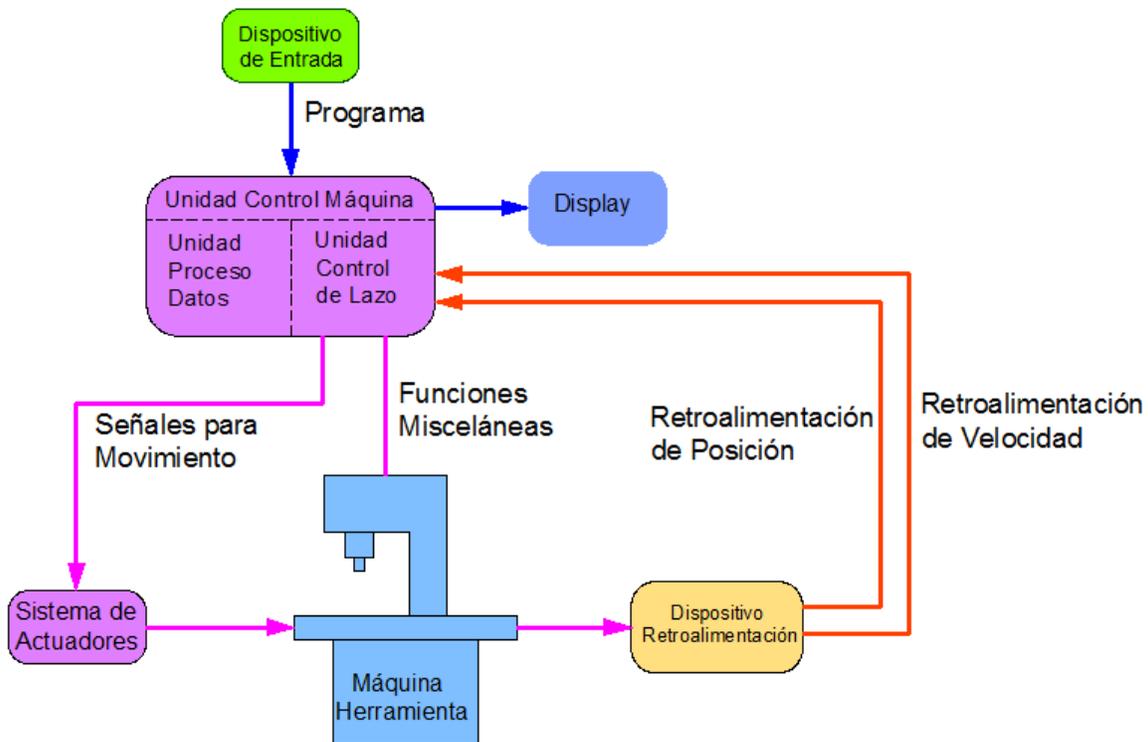


Figura 3.2 Principio de operación de una máquina CNC [30].

Un dispositivo de entrada, el cual es un medio de transferencia de información, conectado directamente a la Unidad de Control de la Máquina (UCM). Hay dos subunidades en la UCM: Unidad de Procesamiento de Datos (UPD) y la Unidad de Control del Lazo (UCL).

La UPD recibe y almacena el programa a ejecutar, primeramente interpreta y decodifica la parte del programa en código máquina, posteriormente calcula las posiciones intermedias en términos de unidades básicas de longitud (BLU). Los datos calculados son transferidos a la UCL.

Los datos procedentes de la UPD se convierten en señales en la UCL para el control de los actuadores del sistema, y así desempeñar los movimientos requeridos de la máquina en cuestión.

Máquina Herramienta, puede ser cualquier tipo de equipo o máquina herramienta. Para obtener alta exactitud y repetibilidad, el diseño de guías así como los tornillos de transmisión son de vital importancia.

Sistema de Actuadores es un componente importante para una máquina CNC, ya que la exactitud y repetibilidad depende mucho de las características y el desempeño de éstos.

El requerimiento es que el sistema de actuadores tenga una buena respuesta de acuerdo a las instrucciones programadas, este sistema normalmente utiliza motores eléctricos. Los motores están acoplados ya sea directamente o a través de una caja de engranajes del tornillo de avance de la máquina. Son tres tipos de motores eléctricos comúnmente utilizados: servomotores, motores de pasos y motores lineales.

Dispositivos de Retroalimentación. Con la finalidad de tener una operación de precisión de la máquina CNC los valores tanto de posición y velocidad de los ejes necesitan ser actualizados constantemente. Los dispositivos de retroalimentación de posición normalmente utilizados son encoder lineal y rotativo.

Encoder lineales. Es un dispositivo montado sobre la mesa de la máquina para medir el desplazamiento actual de la corredera de tal forma que el juego mecánico, motores, etc. no causen ningún error de los datos de realimentación. En la figura 3.3 se ilustra este tipo de encoder lineal.

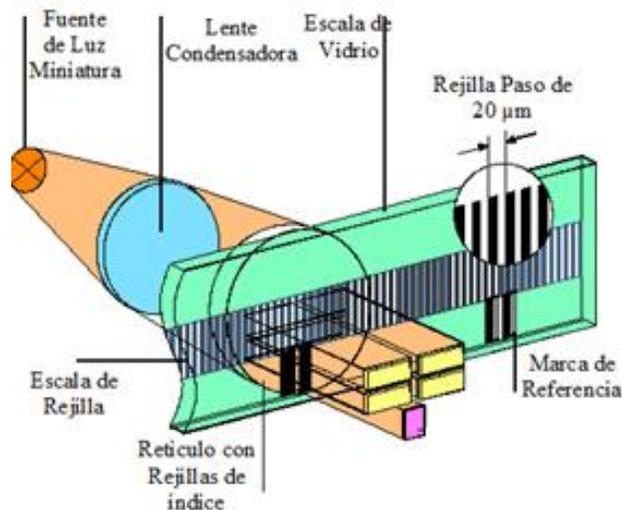


Figura 3.3 Transductor Lineal (Cortesía de Heidenhain) [30].

Encoder Rotatorios. Dispositivo montado en el extremo de la flecha del motor o tornillo de transmisión para medir el desplazamiento angular. En la figura 3.4 se muestra un diagrama de encoder rotatorio.

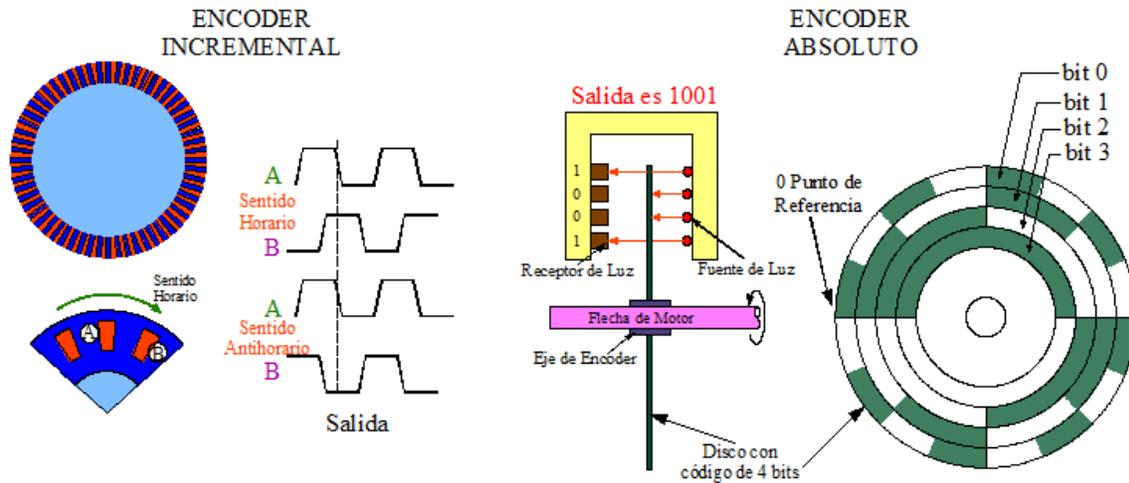


Figura 3.4 Encoder Rotatorio Incremental y Absoluto [30].

Unidad de Visualización o Pantalla. Es un dispositivo de interacción entre la máquina y el operador, cuando la máquina está funcionando la pantalla muestra el estado tal como la posición de la máquina, velocidad de husillo, velocidad de avance, programa de la pieza, etc. En la figura 3.5 se ilustra un ejemplo de interfaz.

En máquinas CNC avanzadas, la pantalla muestra una simulación gráfica de la trayectoria de la herramienta para que el programa de la pieza pueda verificarse antes de ejecutar el maquinado.



Figura 3.5 Pantallas de Visualización [30].

3.2 Técnicas de Control en Sistemas CNC

En un sistema CNC, el programa de control realiza las interpolaciones, controla la velocidad de avance, aceleración y desaceleración, y contiene contadores de posición que muestran la distancia incremental al final del segmento actual a lo largo de la ruta deseada. 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 las trayectorias de maquinado [27]. Estas trayectorias se obtienen generalmente de una combinación de segmentos lineales y circulares, por consiguiente el programa de control debe contener subrutinas para la realización de éstas interpolaciones.

La interpolación es un método mediante el cual, conocidos los valores que toma una función en dos puntos (A, B), se determina con cierto grado de exactitud los valores de un tercer punto (C) comprendido entre A y B.

Existen diferentes métodos de interpolación: lineal, circular, helicoidal, parabólico, curvas de Bézier, B-Splines y B-Splines Racionales No Uniformes (NURBS). Sin embargo todos los controles de máquinas CNC convencionales solo proveen interpolaciones de tipo lineal y circular. Los demás métodos de interpolación son utilizados por la industria que manufactura partes de formas complejas tal como partes de aeronaves y moldes por mencionar algunas.

Básicamente existen dos tipos principales de técnicas de control en los sistemas CNC para generar las señales de referencia a través del programa de control, una es mediante una secuencia de pulsos (técnica por referencia de pulsos) y la otra es con base a palabras binarias (técnica por muestreo de datos) [33].

3.2.1 Técnica por Referencia de Pulsos

En la interpolación por referencia de pulsos, una computadora genera una secuencia de pulsos, como señal de referencia, para el movimiento de cada eje, cada pulso generado produce un movimiento equivalente a una Unidad Básica de Longitud (BLU-Basic Length Unit), o 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 [34, 35]. Estos pulsos pueden accionar un motor a pasos en un sistema de lazo abierto (ver figura 3.6). 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 [34].

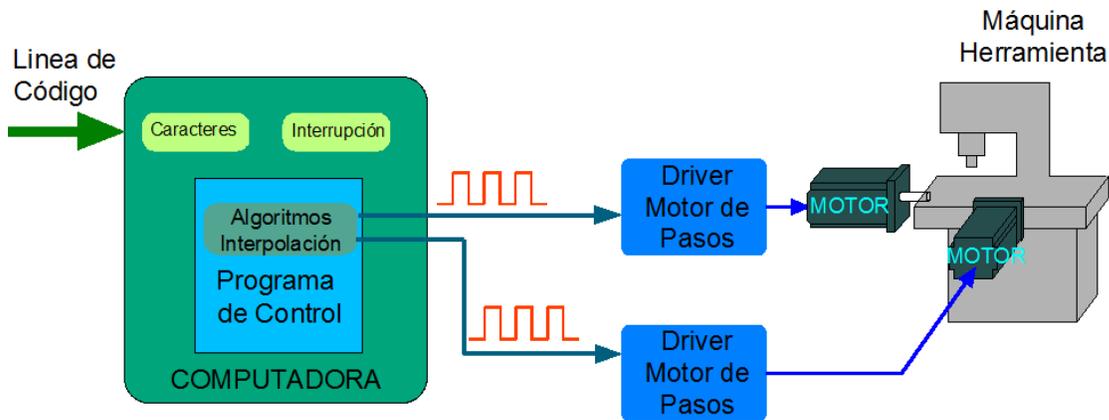


Figura 3.6 Sistema CNC en lazo abierto por referencia de pulsos.

3.2.2 Técnica por Muestreo de Datos

Un sistema CNC, que utiliza este tipo de técnica (ver figura 3.7), 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. Esta señal de error es enviado en intervalos de tiempo fijo a un convertidor digital-analógico (DAC-Digital to Analog Converter), el cual a su vez proporciona un nivel de voltaje proporcional requerido para la velocidad en cada uno de los ejes [36].

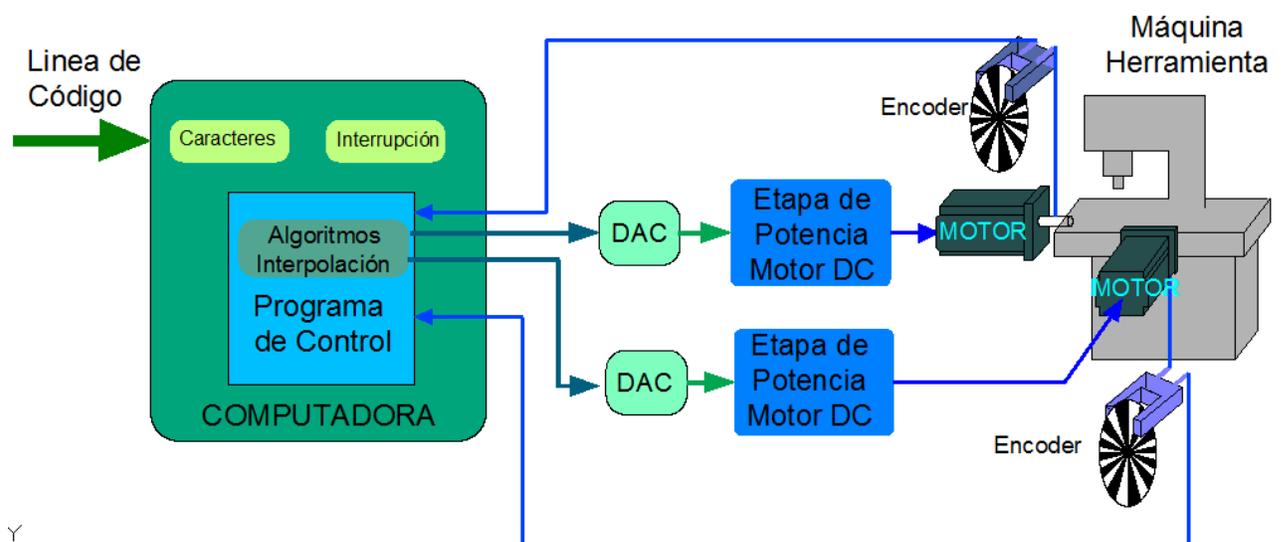


Figura 3.7 Sistema por referencia de palabra.

Ambas técnicas, requieren distintas rutinas de interpolación en el programa de control para generar sus correspondientes señales de referencia. En la tabla 3.1 se resume la comparación entre estas técnicas. En el caso de la *técnica por referencia de pulsos* la velocidad de avance de un eje está sujeta a la velocidad de la Unidad de Procesamiento, pero comparado con el *método de datos muestreados*, se tiene una mejor precisión. El *método de datos muestreado* es útil cuando se requieren altas velocidades en los movimientos. Sin embargo cuando se desea interpolar círculos, lo cual se lleva a cabo mediante la conexión de pequeños segmentos lineales, se presentan errores de cálculo por redondeo, que son acumulativos en cada iteración, lo que trae como consecuencias considerables errores en el contorno de la trayectoria. Además es necesario agregar un complejo control de lazo cerrado y gran capacidad de memoria, comparado con el método por referencia de pulsos.

	Método por Referencia de Pulsos	Método por Datos Muestreados
Descripción	Este método se basa en iteraciones controladas por interrupciones. Cada vez que ocurre una interrupción se ejecuta una iteración de la rutina del programa de control, la cual envía las señales hacia los actuadores.	En este método se basa en que cada cierto periodo de tiempo se calcula los puntos a interpolar y estos datos son transmitidos a cada uno de los ejes.
Ventajas	Ideal para alta precisión de maquinado.	Apto para alta velocidad de maquinado.
Limitaciones	Está limitado para altas velocidades de maquinado.	Inadecuado para alta precisión de maquinado.
Observaciones	Apropiado para controlar motores de pasos. Debido a que la velocidad de los ejes depende de la frecuencia con que se ejecuta el algoritmo de control, se requiere una Unidad de Procesamiento con buen desempeño.	Generalmente se utiliza cuando se controlan motores de corriente directa y servomotores.

Tabla 3.1 Comparación entre las técnicas de control usadas en CNC.

De acuerdo a la comparación de la tabla 3.1 nos impone a elegir la técnica de control por *referencia de pulsos*, tomando en cuenta los requerimientos de

precisión y el tipo de actuadores. En todos los sistemas de control hasta el momento desarrollados en el GMM, se ha optado por esta técnica debido a la simplicidad para ser programados, y su fácil implementación para el tipo de actuadores empleados (motores de 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 [39], pero esto se compensa con una gran precisión en los maquinados.

3.3 Métodos de Interpolación por Referencia de Pulsos

Los métodos de interpolación que emplean esta técnica, están basados en iteraciones controladas por la ejecución de una interrupción, es decir, cada vez que ésta se ejecuta se produce una iteración del método de interpolación, generando un pulso como salida hacia el(los) actuador(es) correspondiente(s) [40]. 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 tres:

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

3.3.1 Interpolación Lineal por DDA

En la figura 3.8 se muestra el diagrama de bloques del procedimiento para realizar una interpolación lineal.

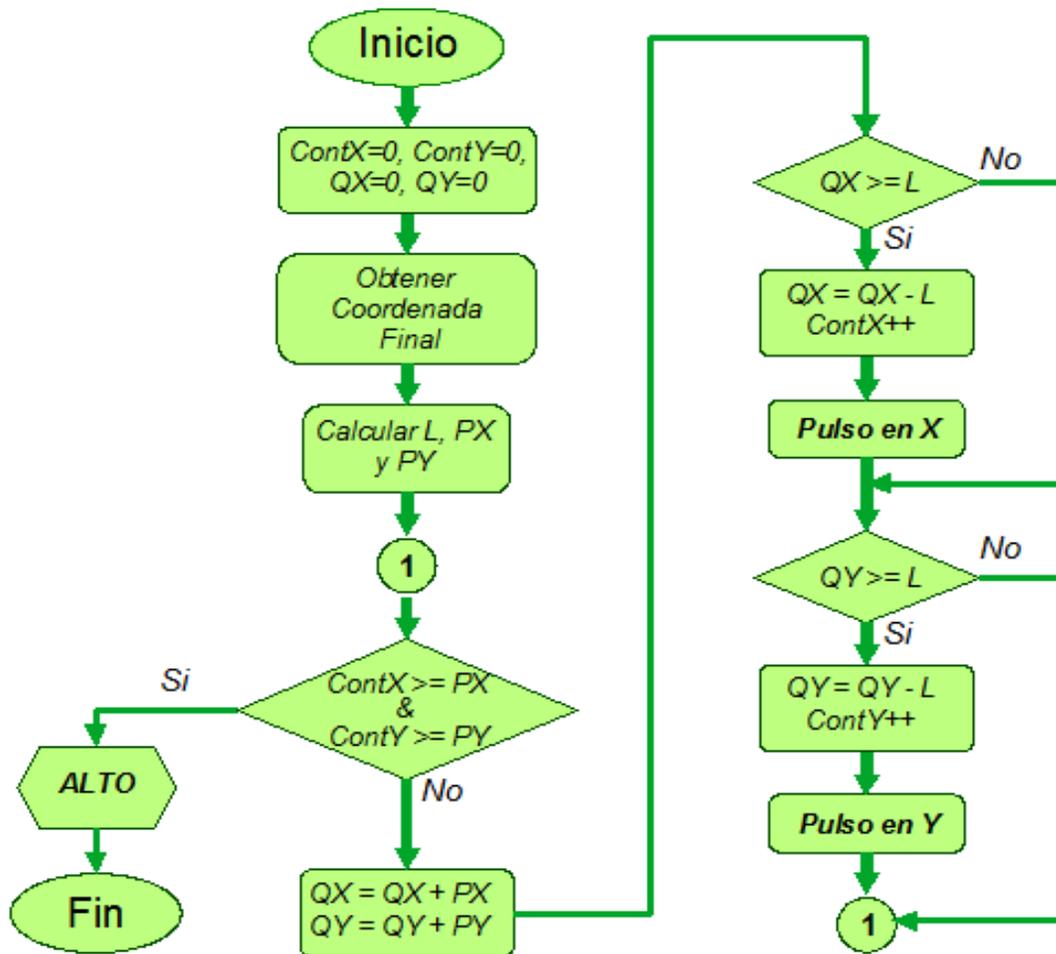


Figura 3.8 Algoritmo DDA para segmentos rectilíneos.

Este método de interpolación, está basado en adiciones, substracciones 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 las variables a y b en la figura 3.9, se almacenan en los registros correspondientes de acuerdo a la proyección del eje coordenado PX y PY .

La variable L es la hipotenusa equivalente al desplazamiento lineal y los registros PX y PY denotan los desplazamientos de los ejes X y Y , equivalente a los valores absolutos a y b . Los valores iniciales de los registros QX y QY son cero.

Durante cada iteración, los valores almacenados en PX y PY son sumados al contenido de otro registro, denotado Q . El desborde de cada uno de estos registros (uno por cada eje coordenado, QX y QY) consiste en sobrepasar la

magnitud de la hipotenusa L (condición de desborde). Además de esto, se verifican los valores de los contadores de pulsos, $ContX$ y $ContY$, cuyo valor de desborde o fin del ciclo de interpolación está determinado por el cociente de las proyecciones entre la resolución del sistema.

Cuando el desborde de los registros QX y QY ocurre, se genera un pulso en la dirección correspondiente donde se originó dicho desborde. Una vez generado el pulso, el valor de L es restado al registro suma QX o QY . De esta forma se actualizará el registro contador de pulsos correspondiente, $ContX$ o $ContY$.

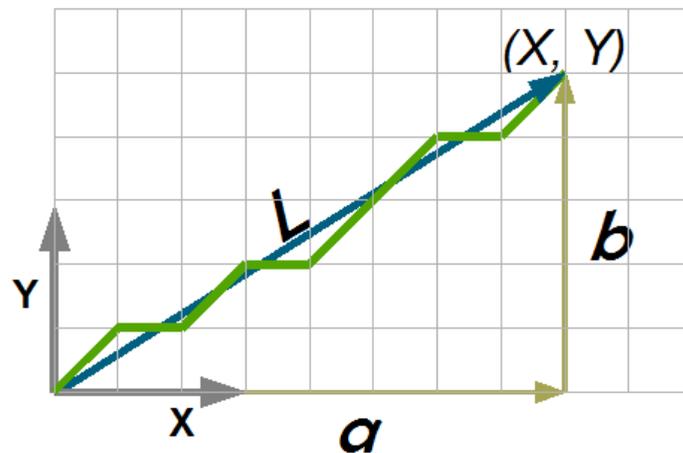


Figura 3.9 Resultado de una interpolación lineal.

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

$$f = \frac{d \times F}{60} \quad (3.1)$$

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

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

F : es el avance de la herramienta durante el maquinado en mm/min. La condición de desborde está dada por:

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

3.3.2 Interpolación Circular por DDA

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 PX y PY cambian al ir avanzando sobre la circunferencia. El desbordamiento ocurre cuando el valor del registro QX o QY excede el radio del arco $R = \sqrt{i^2 + j^2}$, donde i y j son las componentes absolutas en X y Y respectivamente desde el inicio del arco al centro del mismo. Los valores iniciales de PX y PY , son las coordenadas iniciales Y_i y X_i respectivamente. En la figura 3.10 se muestra el diagrama de bloques del procedimiento.

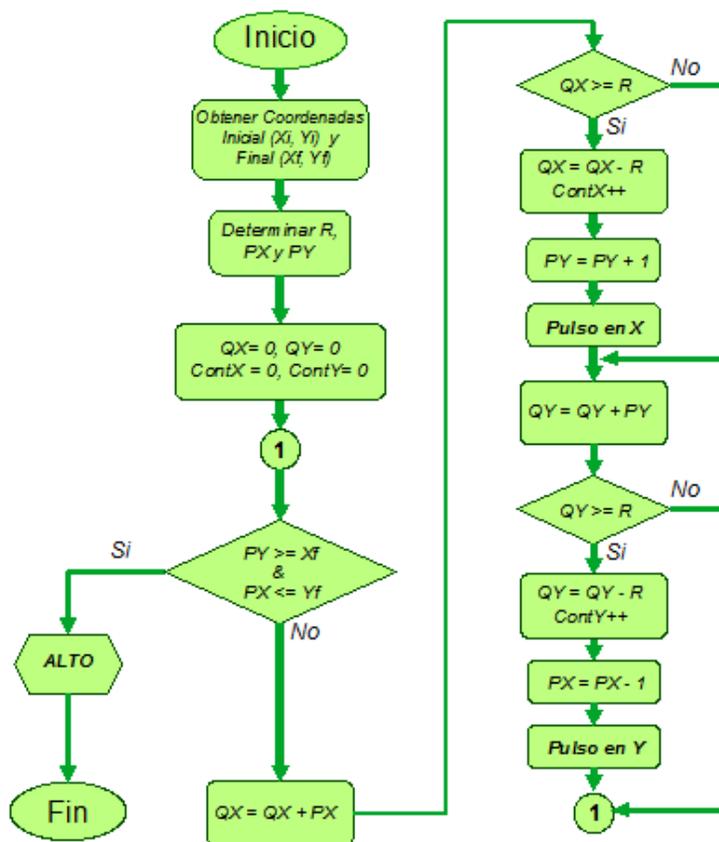


Figura 3.10 Algoritmo DDA para segmentos circulares (1er cuadrante).

Por Ejemplo, considere un arco construido en sentido horario que inicia en el punto (0,10) y termina en (10,0) (cuadrante 1). Para este caso se cargan los valores iniciales $PX = 10$ y $PY = 0$, cada vez que ocurre un desbordamiento, se emite un pulso en la dirección correspondiente donde se originó el desborde, el valor de PX o PY aumenta o disminuye en una unidad, dependiendo el sentido en que se ejecuta la interpolación. En este ejemplo, podemos observar en el

diagrama de flujo de la figura 3.10, que las variables PX y PY se almacenan en registros contrarios. Es decir, dependiendo del cuadrante en el que nos encontremos, al generarse un pulso en la dirección del eje X aumenta en una unidad el registro PY , y si fuese un pulso para el eje Y disminuye en una unidad el registro PX . El fin de la interpolación ocurre hasta que se alcanza el punto final correspondiente a los valores de PY y PX ($X_f = PY, Y_f = PX$). El resultado de la interpolación se ilustra en la figura 3.11.

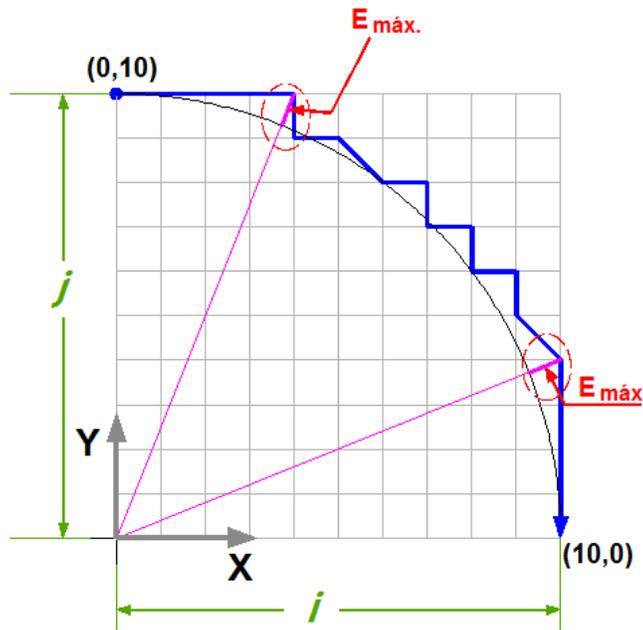


Figura 3.11 Resultado de Interpolación circular usando Algoritmo DDA (1er cuadrante).

El error radial teórico máximo para este algoritmo es aproximadamente 1 unidad de longitud básica (BLU) o resolución, ver figura 3.11.

3.3.3 Interpolación Circular Incremental (o 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_k , la cual es calculada por:

$$D_k = X_k^2 + Y_k^2 - R^2 \quad (3.4)$$

La variable D_k determina la dirección en la cual se emitirá un pulso y el cuadrante en el que se efectuará. Por ejemplo, tomando el mismo segmento circular de la figura 3.19, el algoritmo se ejecuta siguiendo las siguientes reglas:

1. $D_k < 0$: significa que un punto (X_k, Y_k) , está dentro del círculo. En este caso la dirección de movimiento debe ser X positivo.
2. $D_k > 0$: significa que un punto (X_k, Y_k) , está fuera del círculo, y el movimiento debe ser en dirección Y negativo.
3. $D_k = 0$: significa que un punto (X_k, Y_k) , está sobre la circunferencia, y el movimiento puede efectuarse en alguno de los dos casos anteriores.

El procedimiento de este algoritmo se ilustra en el diagrama de la figura 3.12.

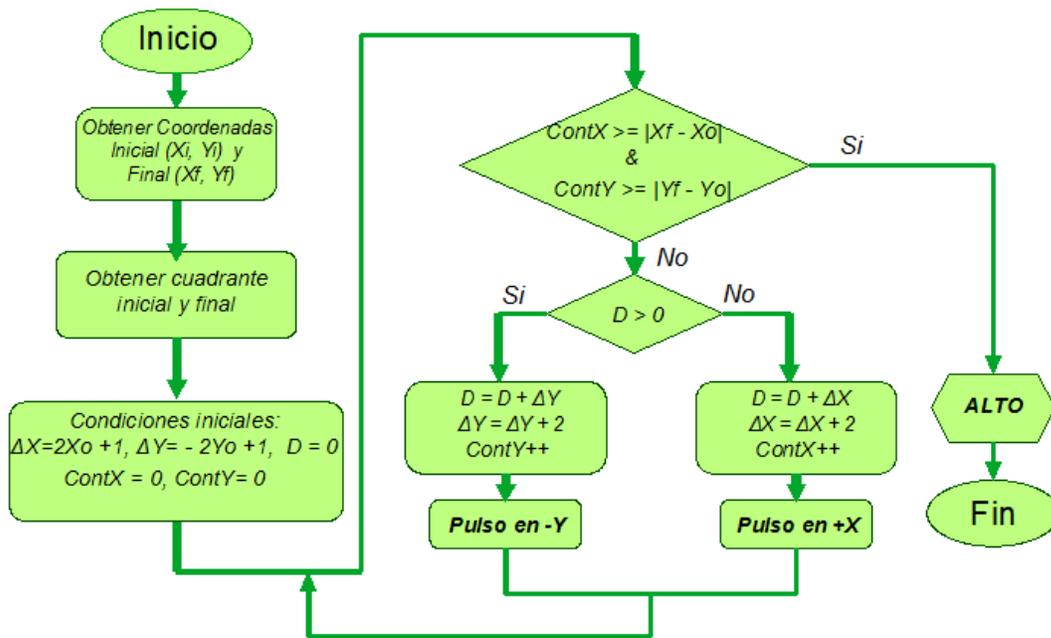


Figura 3.12 Algoritmo de interpolación incremental.

Después de que un pulso se ha efectuado, aplicando las reglas arriba mencionadas, la posición (X_{k+1}, Y_{k+1}) , es actualizada y el procedimiento se repite hasta alcanzar el punto final deseado X_f, Y_f . En la tabla 3.2 se muestran las primeras 10 iteraciones del algoritmo de interpolación incremental para un cuarto del círculo (1er cuadrante) y en la figura 3.13 se ilustra el resultado de la interpolación.

Iteración	Coordenada	$D_k = X_k^2 + Y_k^2 - R^2$	Dirección
1	X= 0, Y= 10	0	+X
2	X= 1, Y= 10	1	-Y
3	X= 1, Y= 9	-18	+X
4	X= 2, Y= 9	-15	+X
5	X= 3, Y= 9	-10	+X
6	X=4, Y= 9	-97	+X
7	X=5, Y= 9	6	-Y
8	X=5, Y= 8	-8	+X
9	X=6, Y= 8	0	+X
10	X=7, Y= 8	13	-Y

Tabla 3.2 Primeras iteraciones del algoritmo de interpolación por escalones.

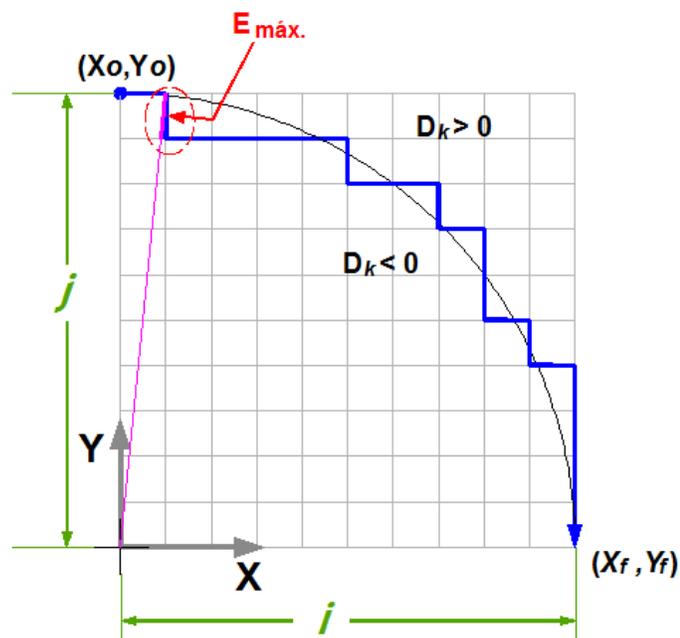


Figura 3.13 Resultado del algoritmo de interpolación por escalones (1er cuadrante).

El error radial teórico máximo para este método es de aproximadamente 1 unidad de longitud básica (BLU), como se observa en la figura 3.21.

Como podemos observar, este algoritmo requiere realizar cálculos que incluyen potencias en cada iteración, esto conlleva a un incremento en el tiempo de procesamiento, aunado a esto es necesario realizar un gran número de iteraciones, el cual se determina con la siguiente ecuación.

$$N = |X_0 - X_f| + |Y_0 - Y_f|, \text{ donde } (X_0 - Y_0): \text{ Es el punto de inicio.}$$

Existe una forma de mejorar la velocidad de procesamiento del algoritmo, esto es modificando y añadiendo un índice más a la expresión 3.4 [38], esto es:

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

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:

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

Estableciéndose un incremento representado por:

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

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

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

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

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

Dónde:

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

y de manera similar para el eje Y , tenemos:

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

Y para:

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

Dónde:

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

Con esto, es posible mejorar 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.

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, 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, una de las desventajas de este método es la incapacidad de originar movimientos simultáneos, esto provoca que más iteraciones (de las realizadas por el método DDA) sean requeridas para describir un círculo.

3.3.4 Interpolación Circular por Búsqueda Directa

Este método aplica la misma estrategia en los cálculos que el método de *interpolación por escalones*, pero con la diferencia de que este método permite el movimiento simultáneo en ambos ejes, lo cual trae como resultado una mejor interpolación del segmento circular [35]. El procedimiento para ejecutar una interpolación circular correspondiente a un cuarto de círculo se ilustra en la figura 3.14.

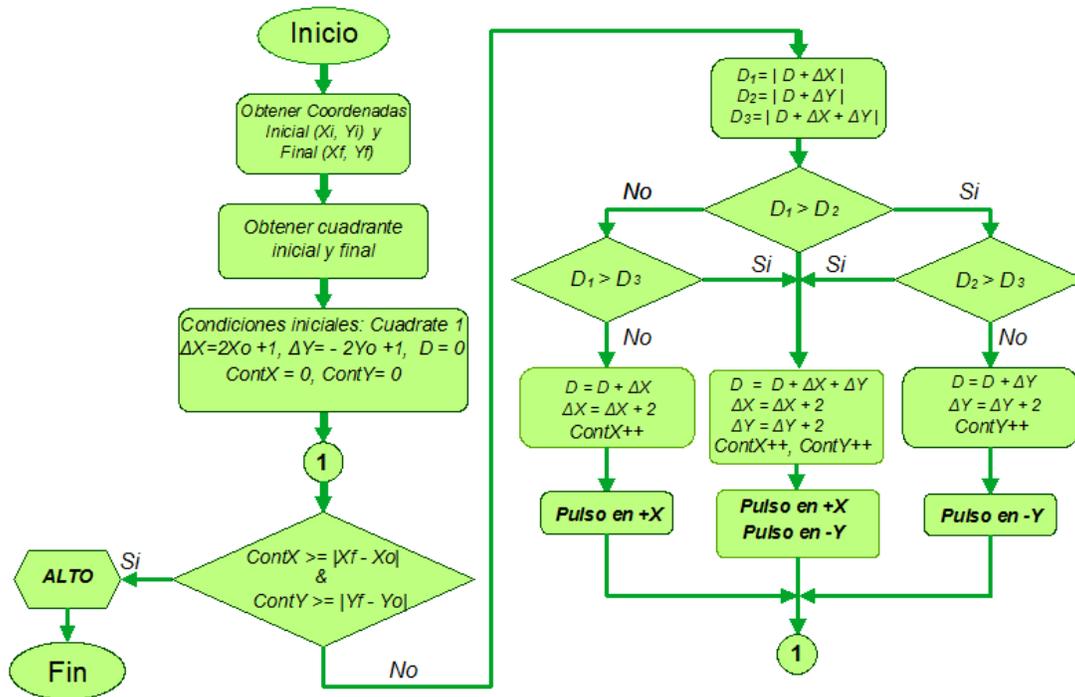


Figura 3.14 Algoritmo por el método de Búsqueda Directa.

El algoritmo por búsqueda directa, como su nombre lo indica busca todos los posibles puntos a los que puede moverse desde la posición actual, y encuentra un punto con el error radial mínimo, evaluando la variable $D_{i,j}$ de la ecuación (3.5).

La elección de estos puntos depende de la dirección de movimiento y del cuadrante de la circunferencia al que esté referida la trayectoria.

Por ejemplo, para un movimiento en sentido horario en el primer cuadrante, se deben considerar tres casos para determinar la ruta del error mínimo:

- 1) Incremento en una unidad (resolución), en la dirección X, teniendo que evaluar las ecuaciones (3.7) y (3.8) para obtener el valor correspondiente $D_{(i+1,j)}$.
- 2) Decremento en una unidad en la dirección Y, obteniendo de las ecuaciones (3.11) y (3.13) el valor $D_{(i,j+1)}$,
- 3) Incremento en una unidad en la dirección X y simultáneamente un decremento en la dirección Y, aplicando la ecuación:

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

La evaluación de $D_{i,j}$ se realiza en cada uno de los casos, y al comparar los tres valores, se escoge el de menor valor absoluto. Una vez hecha la comparación, se genera el pulso en la dirección correspondiente, además se actualizan los valores de ΔX y ΔY , tal como se hace en el método por escalones. El resultado de la interpolación se muestra en la figura 3.15.

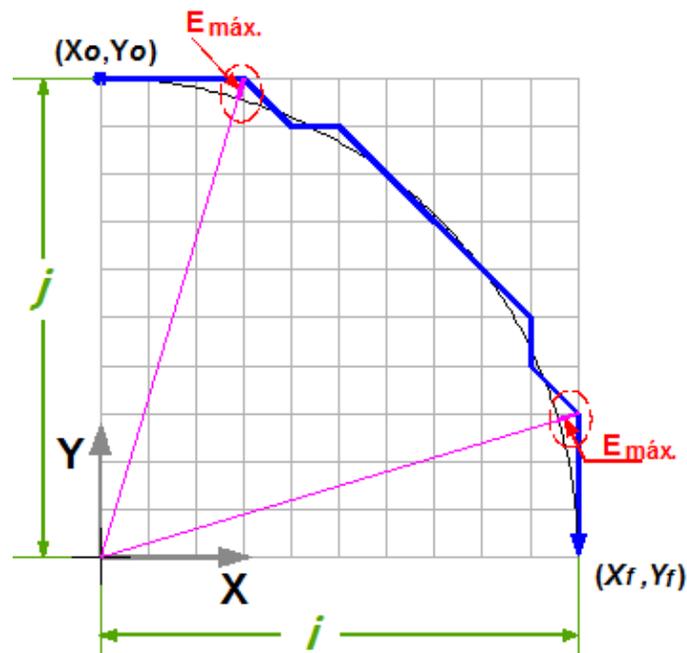


Figura 3.15 Resultado del método de Búsqueda Directa (1er cuadrante).

El error radial teórico máximo en este algoritmo es de $\frac{1}{2}$ unidad de longitud básica (BLU) o resolución, ver figura 3.15. Por lo que la precisión es mucho mejor que el método por escalones.

Algunas características de los tres algoritmos de interpolación descritos se resumen en la tabla 3.3 donde podemos ver, que con el método de *búsqueda directa* el máximo error teórico es de $\frac{1}{2}$ BLU, esto se debe a que en cada iteración se avanza $\sqrt{2} BLU$ para generar la trayectoria y se requieren $N = \sqrt{2}R$ iteraciones para realizar la interpolación de un arco de cuarto de círculo. Este número es aproximadamente 30% menor que el método por escalones y 20% menor comparado con el método DDA. En los otros dos algoritmos el error radial máximo teórico es de $1 BLU$ [34].

Método	Número de Iteraciones N (Para un cuarto de círculo)	$E_{\text{máx}}$
DDA	$\frac{\pi R}{2}$	1
Por Escalones	$2R$	1
Por búsqueda directa	$\sqrt{2} R$	$\frac{1}{2}$

Tabla 3.3 Comparación de los métodos de interpolación.



DESARROLLO DEL SISTEMA DE CONTROL

El GMM se enfoca a desarrollar sus propios sistemas de control en sus prototipos de MMH y busca que sus sistemas cumplan con ciertas especificaciones, como por ejemplo compatibilidad con los CNC comerciales, capacidad de reconocer código G y M generado por algún software CAM, así como también la versatilidad para controlar microequipo con motores de pasos utilizando la técnica de micropasos.

4.1 Características Requeridas del Sistema de Control

- El sistema de control propuesto debe de ser capaz de reconocer las principales instrucciones de código G y M para procesos de fresado.
- Los algoritmos de control deben ser programados en un dispositivo externo a una PC.
- La interfaz gráfica para la comunicación usuario-máquina, deberá desplegarlos parámetros del proceso de manufactura monitoreando cada vez que finalice la ejecución de una línea de código.
- El sistema de potencia debe ser flexible para programarse en el modo de operación de micropasos con el fin proporcionar las señales eléctricas necesarias para el accionamiento de los actuadores.
- La interfaz gráfica deberá ser versátil para poder configurar las características de una MMH, como por ejemplo resolución mínima del sistema por paso de motor y la cantidad de micropasos en su operación.

4.2 Propuesta de control

En el diagrama de la figura 4.1 se muestra la propuesta de control para una MMH, mediante la programación de algoritmos de interpolación programados en un dispositivo externo a la PC, para este caso, se plantea el uso de un Microcontrolador PIC (Peripheral Interface Controller - Controlador de Interfaz Periférico), el cual nos ofrece las posibilidades de un pequeño computador.

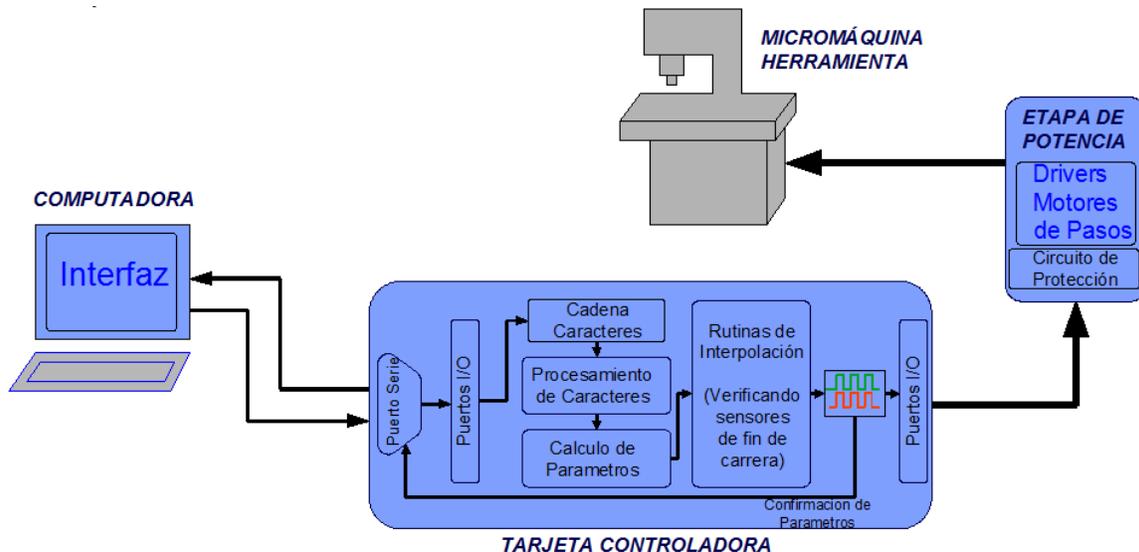


Figura 4.1 Diagrama del Sistema de Control Propuesto.

Como se muestra en la figura 4.1, la tarjeta controladora contará con determinadas funciones, tales como: recepción de cadena de caracteres procedentes de una PC, el procesamiento de las mismas, cálculo de parámetros y ejecución de las rutinas de interpolación (Lineales y Circulares). Esto con la finalidad de proporcionar a la etapa de potencia las señales de control necesarias para el accionamiento de los actuadores de la MMH.

Para determinar los requisitos de entradas/salidas, con que debe contar el microcontrolador, es necesario seleccionar el hardware electrónico que conformará el sistema de control.

4.3 Selección de Hardware Electrónico

Existe una gran diversidad de circuitos de potencia denominados *drivers* para el control de los motores de pasos, capaces de generar una secuencia determinada de pulsos. Estos drivers reciben la información procedente de la tarjeta

controladora y las convierte en señales con el nivel de potencia requerido por los motores, consta fundamentalmente de transistores Mosfet aislados a la tarjeta controladora mediante optoacopladores.

Debido a que el tipo de actuadores que posee la MMH son motores de pasos de dos fases (bobinas) bipolar (dos polaridades), es preciso conocer las formas de generar las secuencias de excitación de las bobinas para su funcionamiento. En la figura 4.2 e ilustra la configuración interna de un motor a pasos bipolar.

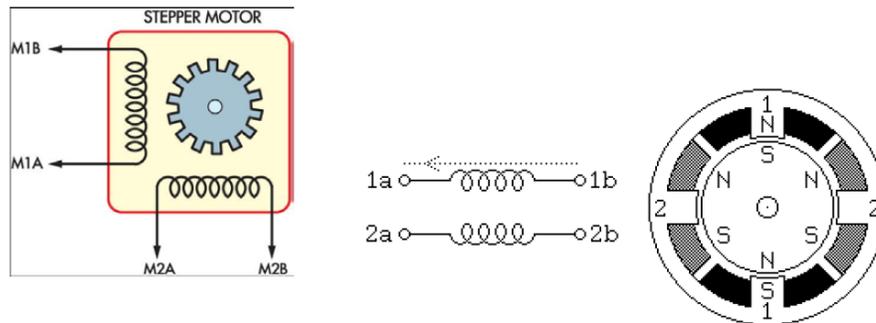


Figura 4.2 Motor de pasos bipolar.

Existen tres modos de operación de los motores de pasos, atendiendo al avance del rotor bajo cada impulso de excitación:

- **Paso completo:** El rotor avanza un paso completo por cada pulso de excitación. La forma de lograr este funcionamiento es energizando una bobina a la vez, ver figura 4.3.

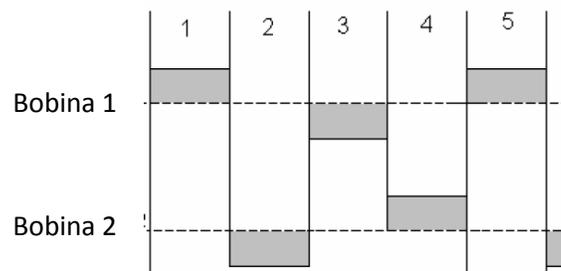


Figura 4.3 Corriente en las bobinas de un motor a pasos.

- **Medio Paso:** Con este modo de funcionamiento el rotor avanza medio paso por cada pulso de excitación, presentando como principal ventaja una mayor resolución de paso, ya que disminuye el avance angular (la mitad que en el modo de paso completo). Para conseguir este modo de

operación, consiste en energizar ambas bobinas antes de avanzar un paso completo, es decir después de energizar una y apagar otra, se prenden las dos al mismo tiempo y a continuación se energiza solo una cumpliendo el desfase de 90° .

- *Micropasos*: Esta forma de control se obtiene haciendo que haya más niveles de corriente por cada una de las bobinas, haciendo que la corriente por una bobina aumente gradualmente hasta llegar a un valor máximo mientras que la corriente por la otra disminuye hasta llegar a cero, ver figura 4.4. De esta forma es posible tener un movimiento más continuo y suave del rotor.

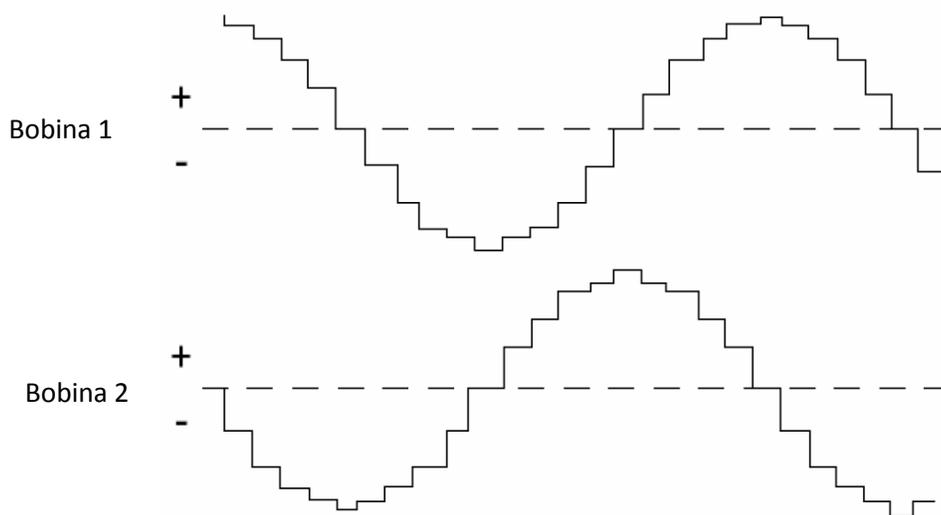


Figura 4.4 Corriente en bobinas en modo micropasos.

El número de micropasos por paso está definido por la cantidad de posibles niveles de control de la corriente. De esta forma también se puede obtener avances de medio paso si se crea un nivel intermedio entre el valor máximo de corriente y cero. Para lograr pasos del mismo valor angular es necesario que el torque se mantenga constante durante todo el ciclo para cada ángulo, esto se logra si la corriente en un embobinado se incrementa mientras la otra disminuye para cada ángulo, manteniendo su suma como fasores siempre constante. Si se aplica una corriente seno por la fase 1 y una coseno por la fase 2, se puede obtener una rotación suave y un posicionamiento preciso bajo condiciones del motor sin carga, además que se obtiene que el motor gire con torque constante para cualquier número de micropasos.

En este trabajo se proponen dos drivers con la capacidad de controlar motores de pasos mediante la técnica de micropasos, estos son: El circuito integrado A3987 del fabricante *Allegro MicroSystems* y el driver R-325 de la compañía *Lin Engineering*.

Para la selección del driver apropiado para nuestro sistema de control y de acuerdo a las características requeridas descritas en la sección 4.1, utilizamos una herramienta cuantitativa denominada *Matriz de Decisión*.

El proceso de evaluación mediante la aplicación de matriz de decisión se describe en los siguientes pasos:

- I. Establecer los factores de mayor importancia (objetivos, rangos o criterios).
- II. Ordenar los factores de manera jerárquica.
- III. Asignar valores relativos a cada factor de acuerdo a su importancia.
- IV. Establecer el rendimiento para cada solución propuesta (calificar la propuesta).
- V. Calcular y comparar en función de los resultados [25], [44].

Para el proceso de evaluación y facilitar la toma de decisiones primeramente describiremos algunas características y modo de funcionamiento de cada uno de los drivers propuestos.

4.3.1 Driver A3987

El A3987 es un circuito capaz de controlar un motor a pasos bipolar en modo de operación de paso completo, medio paso, cuarto de paso y 1/16 de paso (2, 4 y 16 micropasos), con una capacidad de proporcionar una corriente de hasta 1.5 A, con un voltaje de carga desde 7 V a 50 V.

El circuito interno llamado *traductor* es la clave para su fácil implementación en el control de un motor a pasos, simplemente al aplicar un pulso a la entrada STEP del integrado, el motor ejecuta un paso.

La corriente que suministra este circuito es proporcionada por cada uno de los puentes H (transistores DMOS) y regulada a través de un circuito interno PWM, éste usa un circuito disparador que controla el tiempo en que los *drivers* permanecen apagados, internamente este tiempo por default está fijo a 25 μ s.

Para cada puente H, el paso de la corriente se configura con la combinación de un voltaje de referencia externo V_{REF} , una resistencia de sensado R_{SENSE} , y un

voltaje de salida de un convertidor digital analógico (DAC), controlado por la salida del *traductor*. Cuando se aplica una señal a la entrada STEP, el *traductor* automáticamente pasa al siguiente nivel de polaridad y corriente, dependiendo de la resolución de micropasos, configurado con las entradas MS1 Y MS2.

Si el nuevo nivel de salida del DAC es más bajo que el anterior, entonces el *modo de decaimiento de la corriente* para el puente H será en *modo mixto*. Si el nuevo nivel de salida es mayor o igual al anterior entonces el modo de decaimiento será en *modo lento*. Esta selección automática del modo de decaimiento de corriente mejora el desempeño en micropasos, reduciendo la distorsión de las ondas de corriente.

El máximo valor de corriente está limitada a través de la resistencia de sensado (R_{SENSEx}) y de un voltaje de referencia (V_{REF}).

La corriente de carga deseada se calcula con la siguiente expresión.

$$I_{TRIP}(máx) = \frac{V_{REF}}{8} \times R_{SENSE} \quad (4.1)$$

El diagrama interno de éste circuito y su conexión a un motor de pasos es mostrado en la figura 4.5.

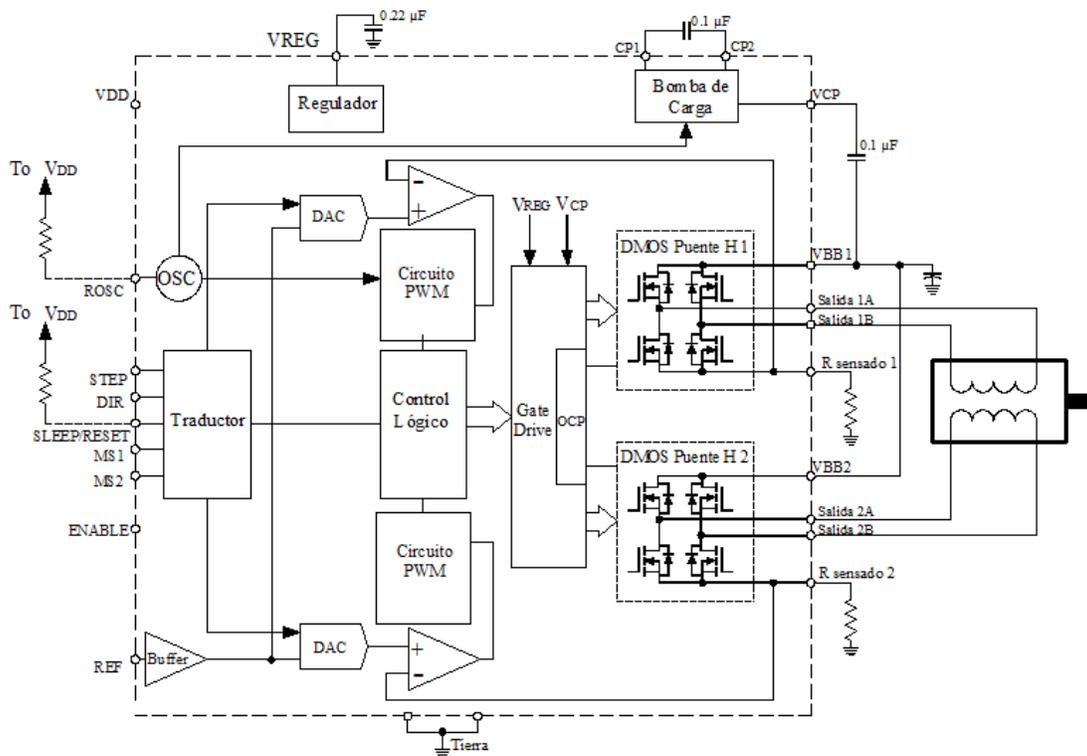


Figura 4.5 Diagrama de bloques funcional del A3987.

En la figura 4.6 se muestra los pines de entrada de este circuito, listo para ser implementado con un microcontrolador.

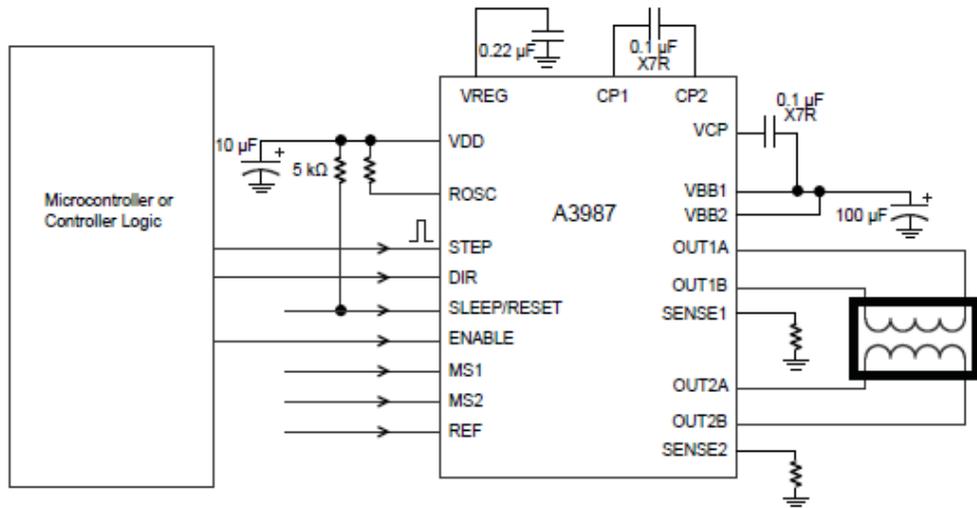


Figura 4.6 Aplicación del A3987.

De la figura 4.6 podemos observar que las entradas del driver A3987, para controlar un motor a pasos con microcontrolador, son las entradas STEP, DIR y ENABLE. A continuación explicaremos el funcionamiento de cada una.

Al aplicar una señal de transición de estado bajo a alto en la entrada lógica STEP, el *traductor* se encarga de generar la secuencia (controla la entrada al DAC y la dirección y flujo de corriente en cada bobina), mediante el cual el motor avanza un paso, es decir que por cada pulso aplicado se produce un paso en el motor, el tamaño de paso está determinado por las entradas MS1 Y MS2.

Con las entradas lógicas MS1 y MS2 se escoge el formato de micropasos (ver tabla 4.1), se recomienda usar resistencias de *pull-up*, con el fin de limitar la corriente de entrada y evitar sobrevoltajes. La resistencia mínima recomendada es de 5 kΩ. En la tabla 4.4 las letras L significa bajo (Low) y H, alto (High).

MS1	MS2	Resolución en Micropasos
L	L	Paso completo
H	L	Medio paso
L	H	Cuarto de paso
H	H	Dieciseisavo de paso.

Tabla 4.1 Configuración para la resolución en micropasos en el circuito A3987.

La entrada lógica *ENABLE*, activa todas las salidas de los FETs (Transistores de Efecto de Campo), cuando está en estado alto lógico las salidas se deshabilitan, y cuando está en cero lógico se habilitan.

El estado de la entrada *DIR*, ya sea alto o bajo, determina el sentido de giro del motor.

El A3987 es un circuito de 26 pines, con dimensiones de 7.8 x 7.4 mm, su encapsulado es TSSOP (Thin- Shrink Small Outline Package), por lo que es necesario elaborar una PCB (Tarjeta de circuito impreso), especial para su montaje.

4.3.2 Driver R-325

Este driver consta de un conjunto de componentes electrónicos, agrupados en una sola tarjeta capaz de controlar un motor de pasos en modo de operación de hasta 256 micropasos, y proporcionar una corriente máxima de 3 A, con un voltaje de carga desde los 15 V a 48 V. En la figura 4.7 se ilustra la tarjeta en cuestión.



Figura 4.7 Driver controlador R-325.

Al igual que el A3987, posee los pines de entrada *STEP*, *DIR* y *ENABLE* para su implementación con microcontrolador, los cuales realizan la misma función al aplicarle una señal lógica de alto o bajo (+5V o 0 V).

Las principales características con que cuenta este driver son las siguientes:

- a. Opera con voltajes desde +15V a 40 V de DC (corriente directa).
- b. Suministro de corriente pico programable desde 0.3 a 3 A.
- c. Corriente de retención programable pico de 0.3 a 3 A.
- d. Resolución programable de 2, 4, 8, 16, 32, 64, 128, 256 micropasos.
- e. Frecuencia máxima de pulsos de 2.5MHz.

4.3.3 Evaluación de los Drivers de Potencia

En la tabla 4.2 se ordenan jerárquicamente los factores importantes para su evaluación, asignando valores de 1 a 3 de acuerdo al nivel de importancia.

Factores de importancia	Valor asignado
Corriente suministrada	3
Voltaje de Operación	3
Resolución de micropasos	3
Hardware externo requerido.	1

Tabla 4.2 Factores importantes en la comparación de drivers.

El resultado de la evaluación de los drivers se describe en la tabla 4.3, los números en letra negritas es el resultado de multiplicar el valor de importancia por la calificación asignada en escala de 0 a 10. La forma de calificar los factores fue de la siguiente manera:

- i. *Corriente suministrada*, se calificó con un 10 para el R-325 ya que la máxima corriente que proporciona es de 3 A útil para cuando se requiere usar motores que consumen este valor de corriente, y para el A3987 se asignó un 6 debido a que suministra máximo 1.5 A.
- ii. Para el *voltaje de operación*, ambos driver se le atribuyó una calificación de 10 por la capacidad de operar con voltajes similares.
- iii. En la *resolución de micropasos*, el R-325 capaz de operar un motor a pasos con mayor resolución de 256 micropasos se evaluó con un 10 y para el A3987 se le asignó un 6 por utilizar una resolución máxima de 16 micropasos.
- iv. En cuanto a *hardware externo requerido* para el control de un motor a pasos, el R-325 no requiere componentes electrónicos adicionales, mientras que el A3987 se precisa de 4 resistencias, 4 capacitores y una PCB para su montaje. Por lo que se les otorgó una calificación de 10 y 8 respectivamente.

Como resultado de la evaluación el *driver a implementar para el sistema de potencia es el R-325*.

Drivers de Potencia	Factores	Corriente		Voltaje		Resolución micropasos		Hardware externo requerido		Total
	Valor	3		3		3		1		
A3987	6	18	10	30	6	18	8	8	74	
R-325	10	30	10	30	10	30	10	10	100	

Tabla 4.3 Resultados de evaluación de los drivers de potencia.

4.3.4 Selección del Modelo de Microcontrolador

Para la elección de un microcontrolador en un diseño en específico se deben considerar varios factores, como por ejemplo herramientas de desarrollo disponibles, precio y lo más importante sus características (gran capacidad de memoria de programa, número de entradas/salidas, protocolos de comunicación, etc.).

Las necesidades de entradas/salidas para nuestro sistema de acuerdo al número de motores a controlar son: seis entradas para los interruptores de límite para sensar el fin de carrera decada uno de los ejes de la máquina y tres salidas para el control de cada driver de potencia.

Existe una gran variedad de fabricantes de microcontroladores, pero para la implementación del sistema de control propuesto, se eligió utilizar una tarjeta con el microcontrolador PIC 18F4550 de la familia Micro Chip. En la figura 4.8 se muestran los componentes de esta tarjeta, y en la tabla 4.4 se enlistan los principales recursos y características que posee. Algunos otros factores que se consideraron en la elección de este microcontrolador son: bastante información disponible que existe para el desarrollo de aplicaciones y compiladores de fácil acceso para su programación.

Características de la tarjeta de control:

1. Alimentación principal.
2. LED testigo de alimentación.
3. Puerto ICSP/MSSP.
4. Pines de conexión/desconexión de puerto USB.
5. Puerto de comunicación R232 principal.

6. Puerto de comunicación USB.
7. Base DIP40 para microcontroladores PIC.
8. Regulador de voltaje.
9. Puerto de comunicación RS232 secundario.
10. Driver para comunicación RS232
11. Botón de Reset.
12. Puertos de entrada/salida.
13. Cristal.

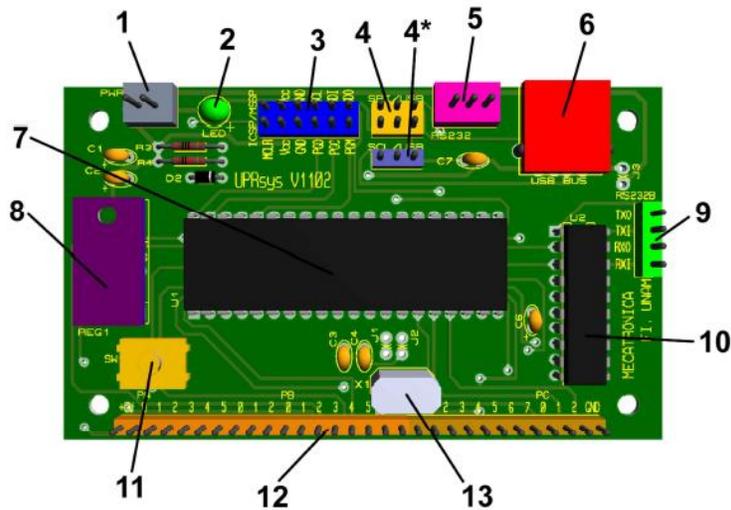


Figura 4.8 Tarjeta de control empleada.

Para la comunicación entre la computadora y la tarjeta de control se decidió emplear el protocolo de comunicación serial RS-232 asíncrono y full dúplex (bidireccional), debido al conocimiento que se posee y su facilidad para ser implementado.

Aunque cualquier PIC con capacidad de comunicación serie UART y con 16 pines de entrada/salida, podía satisfacer el criterio de comunicación y control, la única diferencia destacada es mayor capacidad de memoria de programa, comparado con otros PICs de la gama 18 con el mismo tipo de encapsulado PDIP, lo cual facilita su montaje.

Microcontrolador PIC18F4550	
Tipo de encapsulado	40 pines - PDIP
Frecuencia máxima de operación (MHz).	40
Memoria de Programa (bytes)	32 K
Memoria de Datos (bytes)	2048
Memoria EEPROM (bytes)	256
Fuentes de Interrupción	20
Puertos de entrada/salida	A,B,C,D y E
Pines de entrada/salida	28
Temporizadores	4
Módulos de Comparación, Captura y PWM (CCP)	1
Módulo de comunicación serial	MSSP,USART
Módulo de comunicación USB	1
Módulo de comunicación paralela	PSP
Módulo ADC	10 bits/ 13 canales
Conjunto de instrucciones	75

Tabla 4.4 Características del microcontrolador PIC 18F4550.

Para la programación de los diferentes algoritmos de interpolación, se optó por utilizar el lenguaje de alto nivel y compilador C de CCS (Custom Computer Services Incorporated), y aunque la velocidad de comunicación serial no es una preocupación primaria, la programación en lenguaje ensamblador resultaría mucho más difícil.

4.4 Características del Sistema de Control

En el desarrollo de este sistema de control, se consideran tres aspectos fundamentales: el uso de una PC está limitada para realizar la comunicación entre usuario-máquina a través de una interfaz gráfica, el programa de control será capaz de calcular y ejecutar interpolaciones de tipo lineal y circular, y por último el sistema reconocerá códigos G y M estándar para procesos de fresado.

4.4.1 Modos de operación de la MMH

El sistema de control implementado tiene con cuatro formas de operación para efectuar los movimientos en la MMH:

- 1 Manual (o por *Joystick*).
- 2 Manual por Secuencia de Pasos.
- 3 Por Programa ejecución de Línea por Línea.
- 4 Por Programa ejecución de un programa de manera automática.

En el modo por *Joystick* a través de la interfaz computacional desarrollada, al pulsar un botón etiquetado con el nombre del eje y dirección de movimiento, se envía hacia la tarjeta de control una línea de caracteres correspondientes al desplazamiento deseado, además es posible elegir la velocidad de avance como se considere adecuado.

Para el modo *Por Pasos*, los avances en la MMH se realizan igual manera que en el modo por *Joystick*, cuyos valores son los siguientes: 1 paso que representa el mínimo desplazamiento del sistema, 2, 4, 8, 16, 32, 64, 128, 256, 1024, etc. rutinas que permiten realizar desplazamientos poco críticos en el posicionamiento de la herramienta.

Los otros dos modos de operación que ejecutan un programa de código G de manera automática y línea por línea, son muy parecidos, la única diferencia es que cada vez que se desee ejecutar el siguiente bloque en la modalidad de línea por línea, es necesario presionar el botón EJECUTAR LINEA.

4.4.2 Parámetros monitoreados

Durante la ejecución de cada uno de los modos de operación de la máquina, es necesario que el usuario considere 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.
- Avance (mm/min).
- Cadena de control enviada.
- Interpretación gráfica.

La posición está representada por dos tipos de coordenadas, las que indican 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). El avance se refiere a la velocidad lineal con que se desplaza la herramienta relativa a la pieza de trabajo, la cual está limitada por la velocidad con que se ejecutan los algoritmos de control.

En cuanto al envío y recepción de información, se confirman las líneas de código enviadas entre ambas etapas, la interfaz computacional y la tarjeta de control, dándole al usuario la facilidad de poder detectar posibles fallas de sintaxis o errores en la comunicación. En lo que se refiere a la representación gráfica, esta consiste en graficar las trayectorias representativas del proceso de manufactura.

4.4.3 Códigos G y M reconocidos

La mayoría de los sistemas CNC comerciales maneja un conjunto de aproximadamente 70 instrucciones de código G y 25 instrucciones de código M [36]. La cantidad de instrucciones G y M pueden llegar a variar dependiendo del fabricante del sistema CNC.

Para el caso del sistema propuesto se diseña para que sea capaz de reconocer las instrucciones G y M básicas estándares, para llevar a cabo procesos de fresado en dos y medio ejes, es decir, el desplazamiento de un punto a otro se realiza moviendo tres ejes pero sólo dos de manera simultánea.

Los códigos básicos G y M empleados, para procesos de fresado son:

- 1) G00 Posicionamiento rápido en línea recta a velocidad máxima.
- 2) G01 Interpolación lineal con avance programado.
- 3) G02 Interpolación circular en sentido horario.
- 4) G03 Interpolación circular en sentido anti-horario.
- 5) G04 Tiempo de espera en segundos.
- 6) G28 Retorno automático a posición cero de máquina.
- 7) G54 Selección de coordenadas de trabajo referente al cero pieza.
- 8) G90 Sistema en coordenadas absolutas.

Mientras que la única función M empleada es:

- 1) M30 Fin de programa.

La razón de que nuestro sistema de control, no contenga los códigos M03, M04 y M05, referente al control de giro del husillo de la herramienta de corte, es que este, consiste de un motor neumático, sujetado independientemente a la micro máquina y su accionamiento se realiza de forma manual.

4.4.4 Sintaxis de programación

La sintaxis de cada una de las instrucciones del apartado anterior es la misma que utilizan los sistemas CNC comerciales. Por ejemplo si iniciamos desde la posición cero y queremos realizar una interpolación lineal al punto X500 Y850 con avance controlado de 10 mm/min. Y posteriormente desplazarnos hacia el punto X1000 Y850 lo podemos escribir con cualquiera de las siguientes formas:

- a) G00 X500 Y850 F10;
G00 X1000;
- b) G0 X500 Y500 F10;
X1000;
- c) G0X500Y500F10;
X1000;

4.5 Selección de los Algoritmos de Interpolación

Para el caso de la interpolación lineal, el algoritmo a emplear es el Analizador Diferencial Digital (DDA), descrito en el capítulo 3.

En el caso de la interpolación circular, hasta el momento se han descrito tres métodos de interpolación, y en este apartado, se analizan y evalúan con el fin de seleccionar la mejor opción.

Para la selección del método de interpolación circular más apropiado, utilizamos la misma metodología que en la elección de los drivers de potencia. Aplicando el procedimiento de la *matriz de decisión*, los factores a tomar en cuenta para elegir la mejor alternativa son los siguientes:

1. Mejor aproximación a una circunferencia (determinado por simulación).
2. Menor radial teórico máximo.
3. Numero de iteraciones (Para dibujar una circunferencia).
4. Tiempo en cada iteración.

Para continuar con el proceso de evaluación es necesario realizar una simulación de los diferentes algoritmos de interpolación, por lo que a continuación se describe la simulación de cada uno de estos métodos.

4.5.1 Simulación de los algoritmos de interpolación

Con la finalidad de analizar los diferentes métodos de interpolación circular expuestos en el capítulo anterior, se realizó la simulación mediante software, utilizando el lenguaje de programación Visual C#, la decisión de por qué utilizar este lenguaje es debido a que se tiene mayor conocimiento y experiencia en el desarrollo de aplicaciones.

Para el desarrollo de las simulaciones se realizó una aplicación mostrada en la figura 4.9, en esta aplicación podemos simular círculos de radio no mayor a 40 μm , además un parámetro que podemos cuantificar es el número de iteraciones que ejecuta cada método interpolación.

Para este propósito se consideró que la unidad de longitud básica para dibujar las trayectorias es de 1 μm (resolución). Estas simulaciones son comparadas con una circunferencia ideal de 80 μm de diámetro.

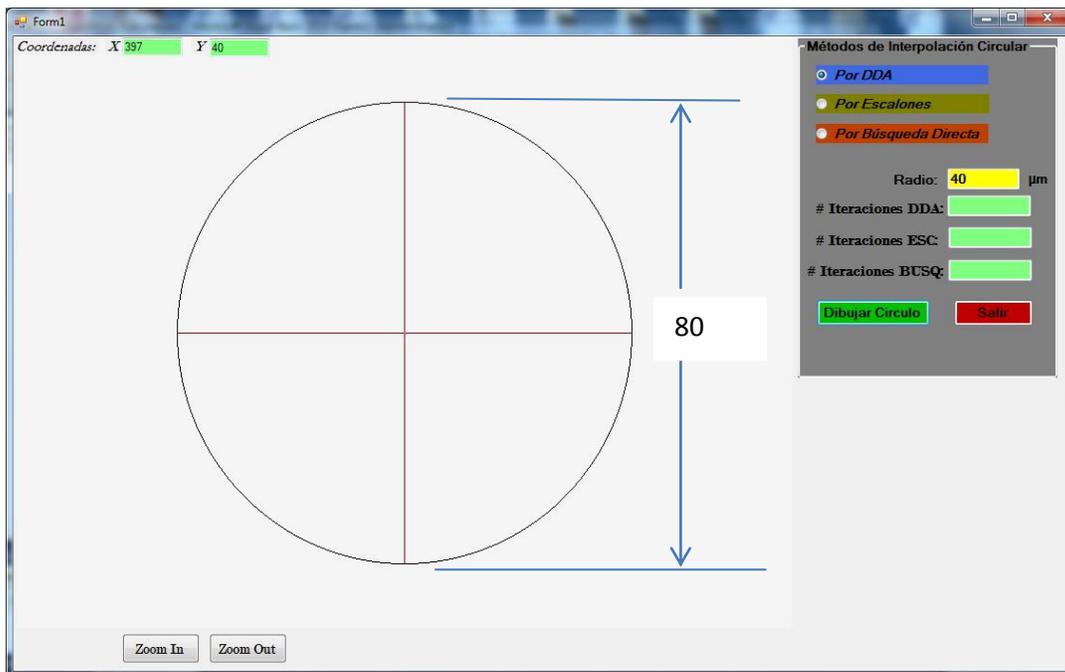


Figura 4.9 Pantalla principal para simulación de los métodos de interpolación.

En la figura 4.10 se aprecia la simulación del método de interpolación por DDA para una circunferencia de 40 μm de radio. Esta representación gráfica en términos generales se puede observar una buena aproximación de la trayectoria circular, esto es debido a que este algoritmo ejecuta pasos simultáneos.

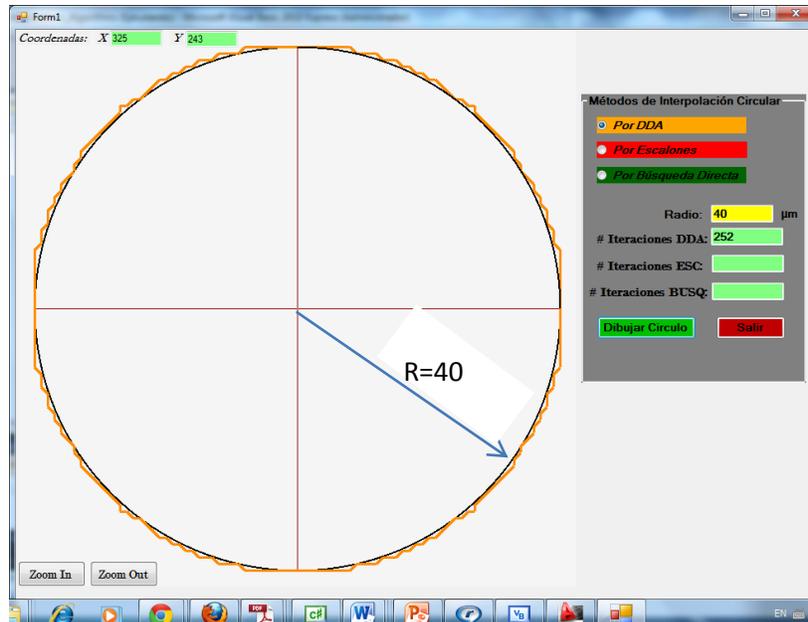


Figura 4.10 Simulación por el método DDA.

Por otra parte, en la figura 4.11 podemos ver la simulación en color rojo correspondiente al método de *interpolación por escalones*, para la misma circunferencia que el método por *DDA*. En esta representación gráfica se aprecia una aproximación *pobre* a una circunferencia ideal, esto se debe a que solo se da un pulso en un único eje por cada iteración.

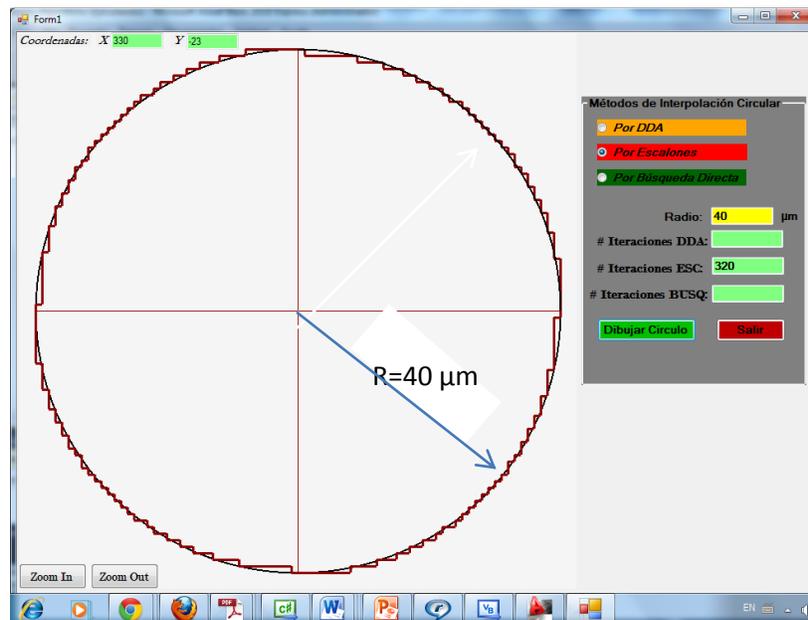


Figura 4.11 Simulación por el método por escalones.

En la tercera simulación mostrada en la figura 4.12 se observa en color verde la trayectoria generada por el método de *interpolación por búsqueda directa*, igualmente para una circunferencia de $40\mu\text{m}$ de radio. En esta representación podemos apreciar una *muy buena* aproximación a la circunferencia ideal, comparado con los dos métodos anteriores. Los resultados de estas tres simulaciones se resumen en la tabla 4.5.

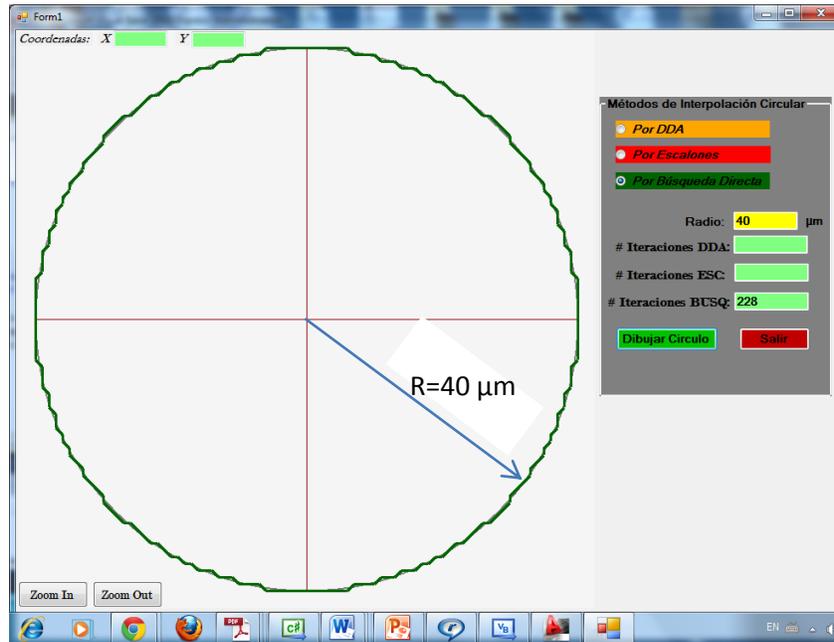


Figura 4.12 Simulación por el método de búsqueda directa.

Por último, en la figura 4.13 se observa la simulación de las tres trayectorias empalmadas, generadas por los tres métodos de interpolación. En esta simulación se aprecia como el método por *búsqueda directa* (color verde) optimiza la trayectoria calculando el error mínimo generado en las tres posibles direcciones dependiendo del cuadrante en que se encuentre. Además en la figura 4.14 se muestra un acercamiento correspondiente a un cuarto de círculo del cuadrante 1 de la figura 4.13.

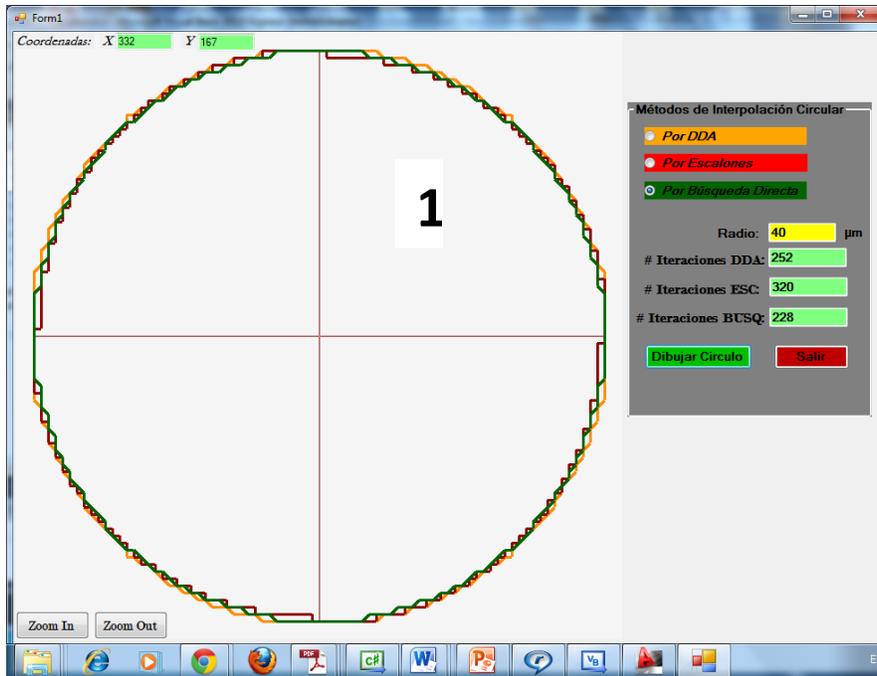


Figura 4.13 Simulación de los tres métodos de interpolación.

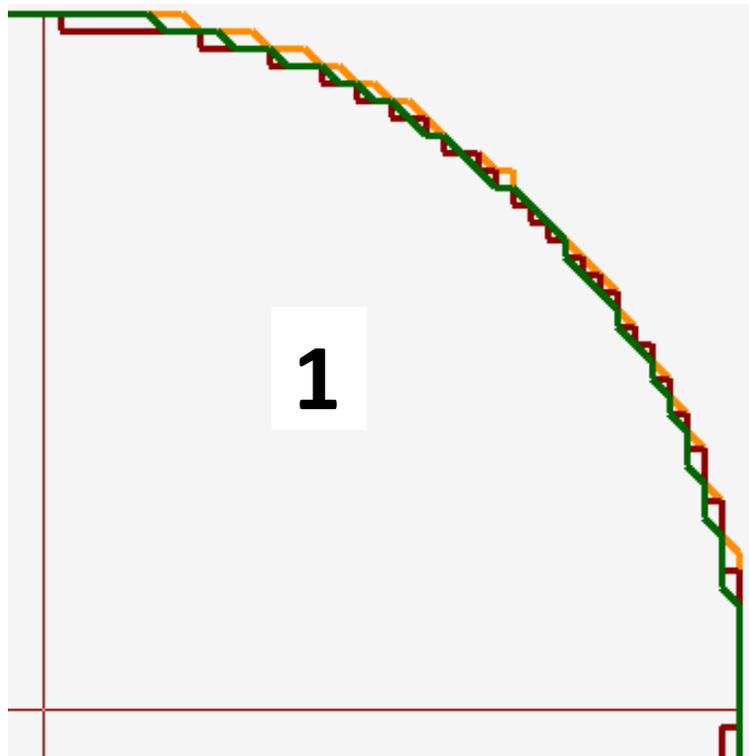


Figura 4.14 Acercamiento de las trayectorias en un cuarto de círculo.

Para determinar el tiempo que toma cada iteración en los diferentes métodos, se desarrolló otra aplicación, ver figura 4.15. Los diferentes algoritmos se programaron para ejecutar un arco correspondiente a un cuarto de círculo. Posteriormente esta información se envía hacia la tarjeta de control, y por medio de un osciloscopio se determina la *frecuencia* a la que se ejecuta el algoritmo y por consiguiente el tiempo que toma cada iteración. Para determinar la frecuencia de los diferentes algoritmos se tomó un radio de 10000 μm . La figura 4.16 nos muestra la frecuencia a la que se ejecuta el algoritmo DDA.

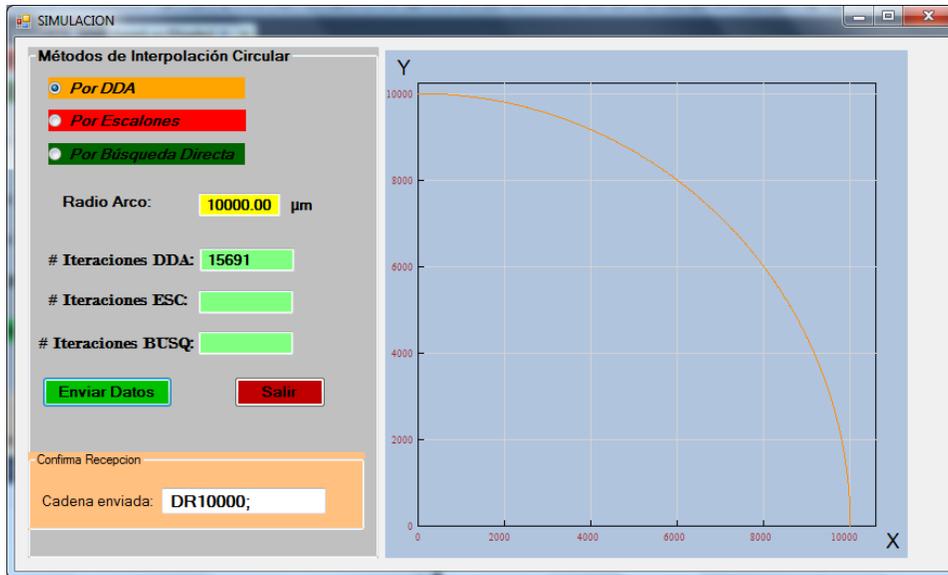


Figura 4.15 Pantalla principal para simulación de la ejecución de un cuarto de círculo.

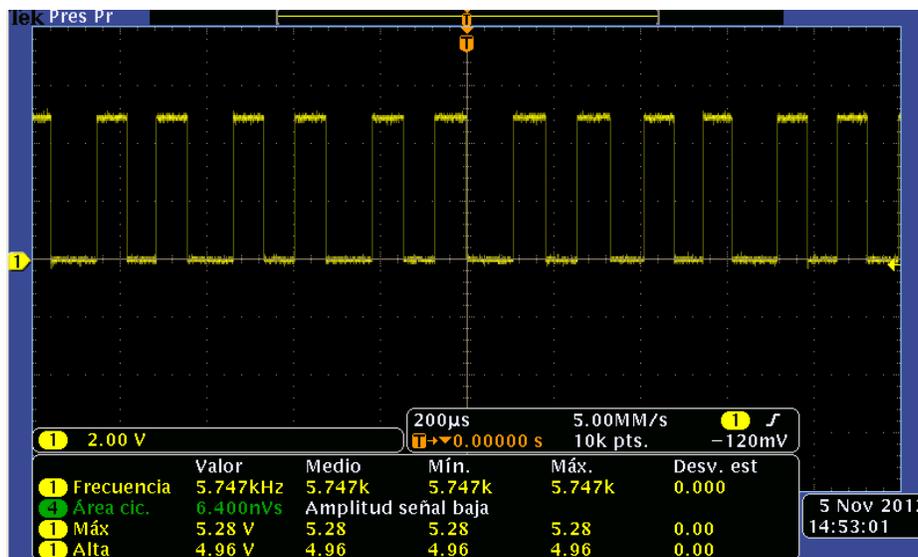


Figura 4.16 Señales generadas por el algoritmo DDA.

De acuerdo con las señales generadas por el *algoritmo DDA* se observa en la figura 4.18 que la frecuencia a la que trabaja este algoritmo es de 5.74 KHz, por lo que el tiempo que toma cada iteración es:

$$t_{DDA} = \frac{1}{f} = \frac{1}{5740} = 174 \mu s$$

El *algoritmo por escalones* se ejecuta a una frecuencia de 8.10 KHz, ver figura 4.19. Con esta frecuencia el tiempo que tarda cada iteración corresponde a:

$$t_{ESC} = \frac{1}{f} = \frac{1}{8100} = 123 \mu s$$

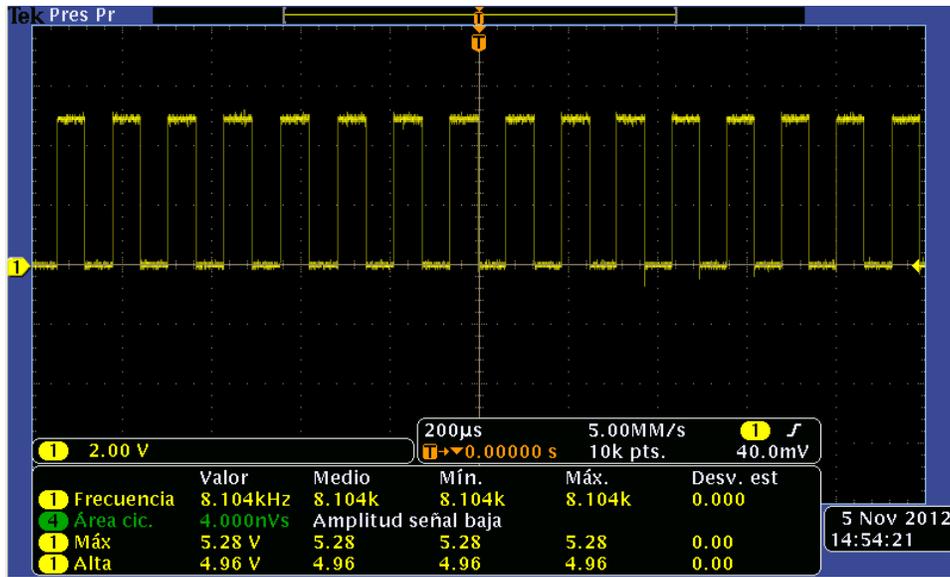


Figura 4.17 Señales generadas por el algoritmo por escalones.

Y en el caso del *algoritmo por búsqueda directa* se desarrolla a una frecuencia de 3.99 KHz, ver figura 4.18, esto corresponde para cada iteración un tiempo de:

$$t_{ESC} = \frac{1}{f} = \frac{1}{3990} = 250 \mu s$$

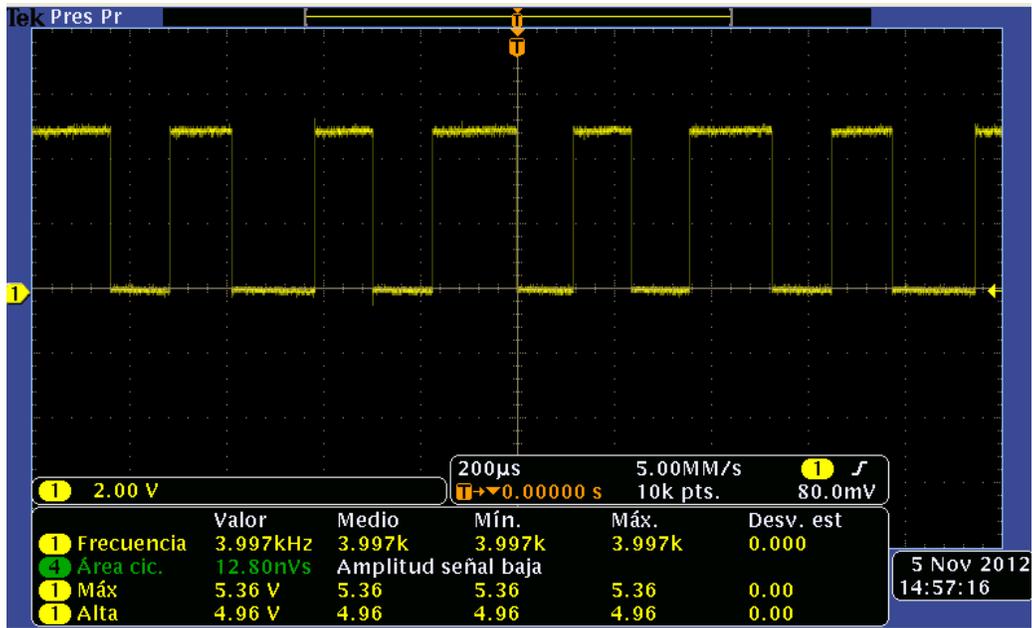


Figura 4.18 Señales generadas por el algoritmo por búsqueda directa.

Posteriormentelos tres métodos de interpolación, son comparados de acuerdo a los resultados obtenidos con la finalidad de tener los argumentos para la elección del método más apropiado y aplicarlo a nuestro sistema de control.

En la tabla 4.5 se establecen los resultados obtenidos de las simulaciones, aquí podemos notar que el *método por escalones* requiere de menor tiempo para ejecutar una iteración, esto es debido a que, al no efectuar pasos simultáneos demanda menor número de comparaciones para decidir la dirección de avance, lo que conduce a consumir menos tiempo en dibujar la trayectoria circular.

Por otro lado, el *algoritmo por búsqueda directa*, en esencia es una mejora del método por escalones en cuanto a número de iteraciones, ya que efectúa pasos simultáneos, pero sin embargo el consumo de tiempo es mayor al registrado por los demás.

Algoritmo	Número de Iteraciones para un círculo.	Error radial teórico máximo [μm]	Tiempo por iteración [μs]
DDA	252	1	174
Por Escalones	320	1	123
Por Búsqueda Directa	228	½	250

Tabla 4.5 Resultados obtenidos de las simulaciones de los algoritmos de interpolación.

4.5.2 Evaluación de los algoritmos de interpolación

Siguiendo con el criterio de evaluación con ayuda de la herramienta *matriz de decisión*, en la tabla 4.6 ordenamos jerárquicamente los factores más relevantes, así como también la asignación de valores relativos de acuerdo a su nivel de importancia. La puntuación de cada valor relativo se califica de la siguiente manera: un 1 significa *poco importante*, un valor de 2 significa *importante* y un valor de 3 *significa muy importante*.

Factores Importantes	Valor relativo asignado
Mejor simulación	3
Error radial teórico máximo	3
Numero de iteraciones (circunferencia)	3
Tiempo por iteración.	1

Tabla 4.6 Factores importantes y su valor asignado.

Para determinar la solución más conveniente de acuerdo a cada factor, se asignan pesos relativos que reflejan nuestra preferencia subjetiva, en una escala del 1 al 10 de la siguiente manera:

- 1) Para el factor de *mejor simulación*, comparando la simulación de los tres algoritmos, se otorga una calificación de 10 a la mejor aproximación a una circunferencia ideal, y de ahí disminuir en una unidad hasta la que presente una pobre aproximación.
- 2) En cuanto al *error radial teórico máximo*, se especifica un valor de 8 para un error de 1 BLU (resolución), y una puntuación de 9 para un error de $\frac{1}{2}$ BLU.
- 3) En relación, al *número de iteraciones*, tomando como referencia al método que realizó menor número de iteraciones, se propuso una calificación de 10, y dividiendo entre 10 el número de iteraciones obtenemos un factor correspondiente al valor de un punto, para ir disminuyendo la calificación.
- 4) Para el caso del *tiempo por iteración*, se fijó un valor de 5 para el método que consumió mayor tiempo, y aplicando la misma regla que en el caso de número de iteraciones, se incrementa la calificación en las demás medidas.

Factores		Mejor Simulación		Error radial teórico máximo		Numero de Iteraciones (Circunferencia)		Tiempo por iteración		Total
Método de Interpolación	Valor relativo	3		3		3		1		
DDA		9	27	8	24	9	27	6	6	84
Por Escalones		8	24	8	24	6	18	10	10	76
Búsqueda Directa		10	30	9	27	10	30	5	5	92

Tabla 4.7 Evaluación de los algoritmos de interpolación.

Finalizando con base a los resultados obtenidos en la tabla 4.7, la opción resultante implica utilizar el algoritmo por *búsqueda directa* ya que representa una mayor puntuación con 92 puntos.

4.6 Desarrollo de la Interfaz Gráfica

Con el objetivo de establecer una comunicación entre el usuario y la MMH, se realizó una interfaz gráfica con el software Visual Studio C#, ya que éste brinda la combinación de potencia tanto de Visual Basic y C estándar por su facilidad y forma estructurada de programación, bajo el paradigma de la programación orientada a objetos.

La interfaz gráfica provee los elementos de interacción necesarios para que el usuario pueda controlar los movimientos de la máquina. Esta interfaz consta de tres ventanas principales: una relativa a la selección del modo de operación y características de la máquina tales como resolución mínima de la micromáquina y cantidad de micropasos a la que trabajarán los motores (ver figura 4.19), y las otras dos están dedicadas a la operación en modo manual y automático.

En la pantalla de la figura 4.19, se presenta al usuario tres opciones que corresponden a las formas de manipular a la MMH, *Modo Manual o Modo Programa y la otra pertenece a la configuración*, en donde el usuario debe ingresar la resolución y cantidad micropasos. Para garantizar que la configuración fue enviada correctamente a la tarjeta de control, los datos elegidos aparecen en el cuadro de texto *Confirmación Config.*



Figura 4.19 Pantalla de inicio para elegir el modo de operación.

En la pantalla mostrada en la figura 4.20 se muestran cinco secciones. En la sección número 1, tenemos un cuadro referente a las modalidades en las que puede operar la micromáquina, por joystick y por pasos, la selección de una de estas opciones habilitará los cuadros correspondientes a estas modalidades de la sección 3.



Figura 4.20 Pantalla para operar en modo manual.

En la sección 2 se muestran las coordenadas relativas al cero máquina (X_m, Y_m, Z_m), al cero pieza (X_p, Y_p, Z_p), a la velocidad de avance y rpm del husillo. La sección 3 le indica al usuario en que eje y dirección se moverá la máquina al pulsar uno de los botones correspondientes, de acuerdo a la distancia en μm o número de pasos seleccionados de un cuadro de lista, y también es posible escoger mediante un cursor la velocidad de desplazamiento desde 1 hasta 12 mm/min.

La sección 4 contiene cinco botones, uno está dedicado para llevar a la máquina a su origen mediante el botón *Home*, otro es para detener cualquier movimiento que se esté ejecutando mediante el botón *Alto*, al pulsar el *WorkPiece*, aparece una ventana para establecer las coordenadas del cero pieza (ver figura 4.12), con el botón *ProgramaAut*. Se abre la ventana para ejecutar un programa de manera automática y con el botón *Salir* salimos de esta aplicación.

Por último en la sección 5, se muestran unos cuadros verdes que indican el estado de los interruptores de fin de carrera de cada uno de los ejes, también se muestra el estado del sensor de contacto de herramienta de corte con la pieza de trabajo. Si están activados cambian a color rojo, en caso contrario permanecen en color verde.

La figura 4.21, muestra la ventana para que el usuario introduzca las coordenadas correspondientes al cero pieza, al presionar el botón *Guardar*, los valores de las coordenadas (G54 o G55) se envían a la tarjeta de control para almacenarlos, ésta nos los regresa, y si el envío fue recibido y guardado correctamente aparecen los datos en el cuadro de color verde *Confirmación de Cero Pza Almacenado*.

The screenshot shows a software window titled "INGRESO DE DATOS MANUAL". The main area is divided into several sections:

- ESTABLECER CERO PIEZA:** This section contains two columns of input fields for G54 and G55. Each column has three rows for Xcp, Ycp, and Zcp, each with a text input field and a value of "0.00 μm ". Below these are two buttons: "Guardar" (green) and "Salir" (orange).
- POSICIÓN MAQUINA:** Located on the right side, it has three input fields for X_m, Y_m, and Z_m, each showing "0 μm ".
- POSICIÓN PIEZA:** Also on the right side, it has three input fields for X_p, Y_p, and Z_p, each showing "0 μm ".
- Confirmation Section:** At the bottom left, there is a green background area titled "Confirmación de Cero Pza Almacenado". It includes a text field for "Cadena Enviada:" and three input fields for Xcp, Ycp, and Zcp.

Figura 4.21 Pantalla para establecer el cero pieza.

En la ventana de la figura 4.22 podemos observar siete partes correspondientes al modo de operación por programa. En la sección 1 podemos seleccionar la forma en que deseamos ejecutar un programa de códigos G, ya sea programa automático, línea por línea o ejecutar la simulación del mismo. Cuando se desea ejecutar el programa en forma automática se habilita el botón *EJECUTAR PROGRAMA AUT.* Para el caso de línea por línea se habilita el botón *EJECUTAR LINEA*, y para la simulación se habilita el cuadro correspondiente.

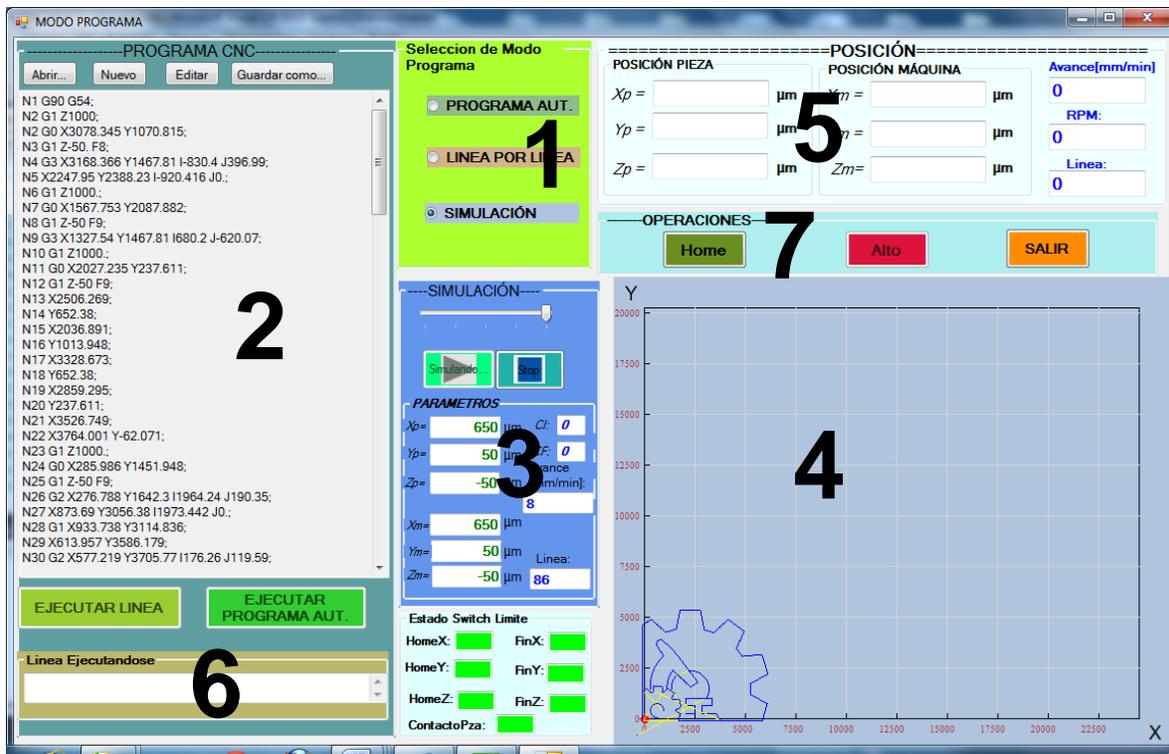


Figura 4.22. Pantalla para operar en modo programa.

En la sección 2 se tiene un cuadro de texto en el que se introduce un programa CNC con líneas de código siguiendo una determinada sintaxis la cual se explicó en la sección 4.4.4. Cuenta con la opción de abrir un archivo de texto que contiene el programa de una pieza generado por un software CAM. Entre otras de las opciones se encuentran la de editar, guardar y crear un nuevo programa.

La sección 3 se ocupa de llevar a cabo la simulación del programa contenido en la sección 2, con el fin de detectar posibles errores en la sintaxis de la programación.

En la sección 4 se representan gráficamente las trayectorias del programa que se está ejecutando con cualquiera de las tres formas elegidas.

Para la *sección 5* al igual que en la pantalla en modo manual se muestran las coordenadas relativas al cero máquina (X_m, Y_m, Z_m), al cero pieza (X_p, Y_p, Z_p), además de la velocidad de avance y rpm del husillo.

La *sección 6* dispone de una casilla donde se despliega el número de bloque o línea de código que se está ejecutando. Y por último la *sección 7* contiene 3 botones, uno para regresar a cero máquina (botón Home, está activo cuando no se está ejecutando un programa), otro es para detener cualquier movimiento que esté realizando la máquina (botón Alto) y el otro es para salir de la modalidad programa.

4.7 Estructura del Programa de Control

El procedimiento que emplea el *programa de control* para desempeñar las funciones de mando en los diferentes modos de operación de la MMH, se ilustra en el diagrama de flujo de la figura 4.23. El código fuente de este programa, se realizó en lenguaje de alto nivel utilizando C de CCS para PIC de Microchip, y grabado en el microcontrolador PIC 18F4550.

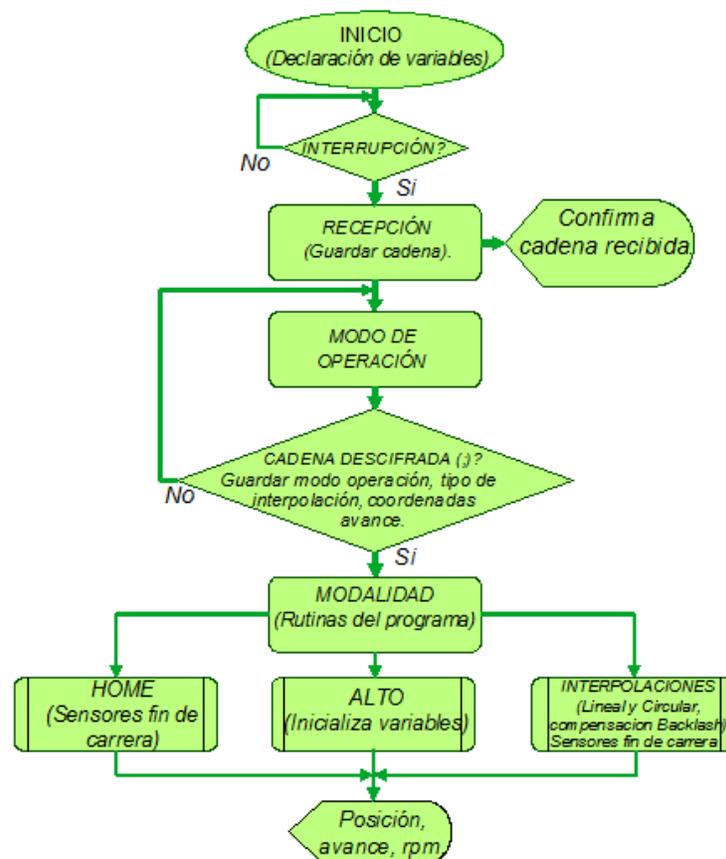


Figura 4.23 Rutinas y subrutinas principales del programa de control.

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 PC. Para verificar 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 llamada *Confirmación de Cadena Enviada para el modo en operación manual y Línea Ejecutándose* en el caso de modo programa.

Una vez recibida la cadena completa, el bloque *MODO DE OPERACION*, analiza carácter por carácter la cadena recibida, esto se lleva a cabo dentro de un *ciclo while* donde se guardan las diferentes variables tales como: modo de operación (manual o programa), tipo de interpolación, las coordenadas, avance, etc. Con base a los parámetros almacenados. Son 6 posibles caminos a seguir en este programa de control:

- 1) Rutina de *alto*.
- 2) Rutina para avanzar por *joystick uno* o dos ejes.
- 3) Rutina para avanzar con una secuencia de *pasos*.
- 4) Rutina para interpretar una línea de código *G*.
- 5) Rutina para interpretar una línea de código *M*.
- 6) Rutina para desplazar los ejes al origen de máquina (*Home*).

Los últimos tres bloques se refieren a todas las subrutinas arriba mencionadas.

4.8 Ejemplos de Operación del Programa de Control

Para describir la forma de procesar una cadena de caracteres de código mediante este sistema, se realizan dos ejemplos, estos ejemplos son propuestos debido a que representan los cuatro modos de operación del sistema. La *Cadena1* ejemplifica los modos *joystick* y por *pasos* (modos similares), mientras que la *Cadena2* ejemplifica los modos *línea por línea* y por *programa* (modos similares). Las cadenas propuestas son las siguientes:

- 1) *Cadena1*: jX+512F8;
- 2) *Cadena2*: G2 X157.33 Y1034.95 I1348.47 J271.89 F10;

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

4.8.1 Ejemplo (1)

Se selecciona la forma de operación manual para operar la máquina y se escoge el modo *Joystick*. Antes de enviar cualquier cadena de control, se tiene la opción de ejecutar o no la rutina de *HOME*, en caso de no hacerlo, el programa de control toma como el *cero máquina* la posición actual en que se encuentren los carros de la MMH y no el cero definido por los interruptores de límite.

Posteriormente se debe elegir la distancia a recorrer en μm de un cuadro de lista, también se elige por medio de un selector variable la velocidad con que se desea avanzar, luego se da *click* al botón que activa el movimiento del eje X, X+, produciéndose el envío de la cadena "jX+512 F8;".

La forma de procesar la *Cadena1* almacenada, se muestra en el diagrama de la figura 4.24.

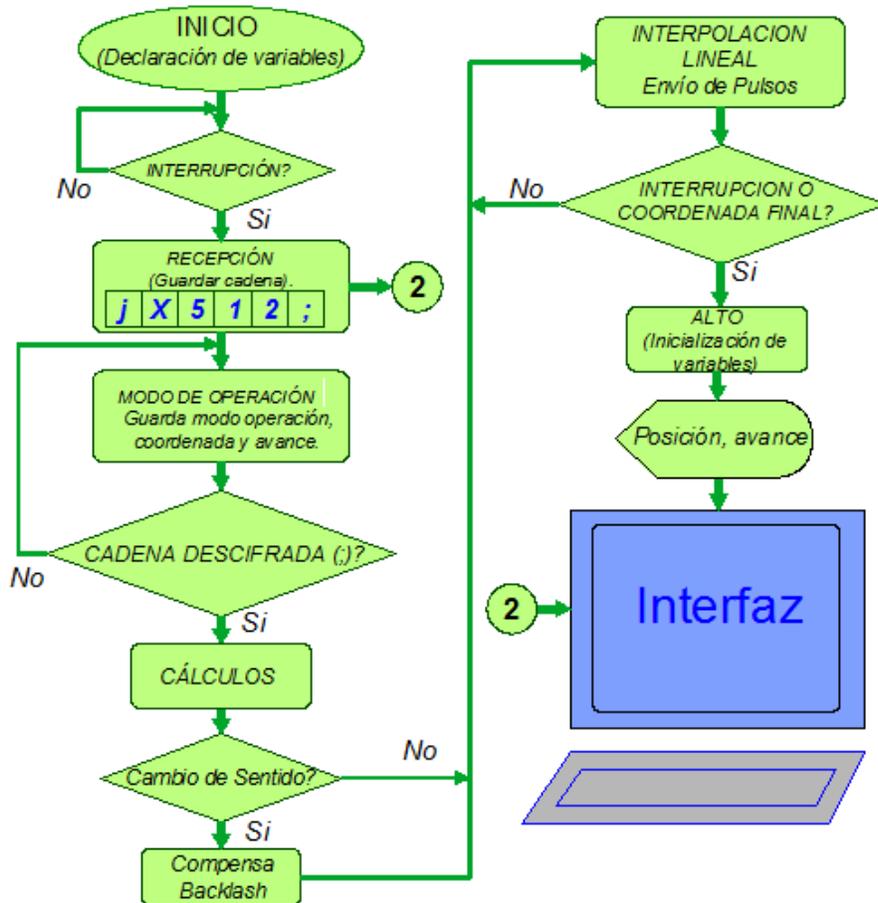


Figura 4.24. Proceso de control para la cadena "jX512F8"

Ahora bien, una vez enviada la Cadena1 (jX+512F8), ésta es recibida por la tarjeta de control en la cual se atiende una interrupción, almacenando el conjunto de caracteres en un vector llamado *LineaGM*. El primer carácter de la cadena se guarda en el elemento cero del vector, el segundo en el elemento uno y así sucesivamente. Al momento de detectar el carácter fin de línea (;) se termina el proceso de almacenamiento y se habilita una bandera llamada *bandera* para indicar que ha finalizado la recepción.

Para decodificar la cadena de control, se ejecuta la rutina de *modo de operación*, entonces se lee carácter por carácter la cadena almacenada, descifra el modo en el cual se quiere operar la MMH. En este caso, el primer carácter letra *j* indica que los desplazamientos se harán por medio de un *joystick*. Posteriormente pasa al siguiente carácter que corresponde a la letra *X*, en este caso se habilita una bandera *ejeX*, para indicar que el movimiento va ser en el eje X.

Posteriormente se ejecuta un ciclo *while* para descifrar los caracteres subsecuentes a la letra *X* los cuales se almacenan en una variable *Xdda* hasta encontrar alguno de los siguientes caracteres: *espacio en blanco, letra Y, F, I o J*. Al término de este ciclo, los caracteres almacenados en *Xdda* se convierten a un número de punto flotante para que más adelante se realicen operaciones matemáticas. El proceso de descifrado continua en la misma línea y se aplica el mismo proceso en caso de encontrar alguna de las letras mencionadas. Este proceso se repite hasta topar con el carácter de fin de línea (;) con lo cual finaliza la rutina.

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.

Una vez que es descifrada la modalidad de control y conocido todos los parámetros correspondiente al tipo de movimiento, tanto como dirección, sentido, avance y desplazamiento en μm , se limpia el vector *LineaGM* para una nueva recepción, e inmediatamente se pregunta si existe un *cambio de sentido*, en relación al movimiento anterior. La comparación se lleva a cabo mediante el estado de dos banderas: *sentidoactual* y *sentidoanterior*, si el estado de estas dos banderas resulta ser diferente entonces se ejecuta una rutina que compensa el juego mecánico. En seguida se procede con la ejecución de la *subrutina de interpolación* enviando una combinación de señales a través de los puertos del microcontrolador hacia los drivers en sus entradas *STEP*, *DIR* y *ENABLE* descritas anteriormente, esta combinación se mantiene grabada durante cierto periodo de tiempo de acuerdo a la frecuencia con que se desea realizar los movimientos de los ejes de la MMH. Dentro de la subrutina de interpolación se verifica

periódicamente el estado de la bandera de recepción. En caso de estar activada se detiene cualquier mediante la rutina *Alto*, se inicializan las variables y se envía la información de la posición de la MMH a la interfaz gráfica. Después se está listo para ejecutarse nuevamente la rutina de recepción.

4.8.2 Ejemplo (2)

Para este ejemplo se deberá escoger la forma de operación *por programa*. Después se introduce en el cuadro de texto *PROGRAMA CNC* la Cadena2 (G2 X157.33 Y1034.95 I1348.47 J271.89 F10;). Enseguida se da *click* al botón *EJECUTAR PROGRAMA AUT.* produciéndose el envío serial de la Cadena2. El procedimiento se ilustra en el diagrama de la figura 4.25

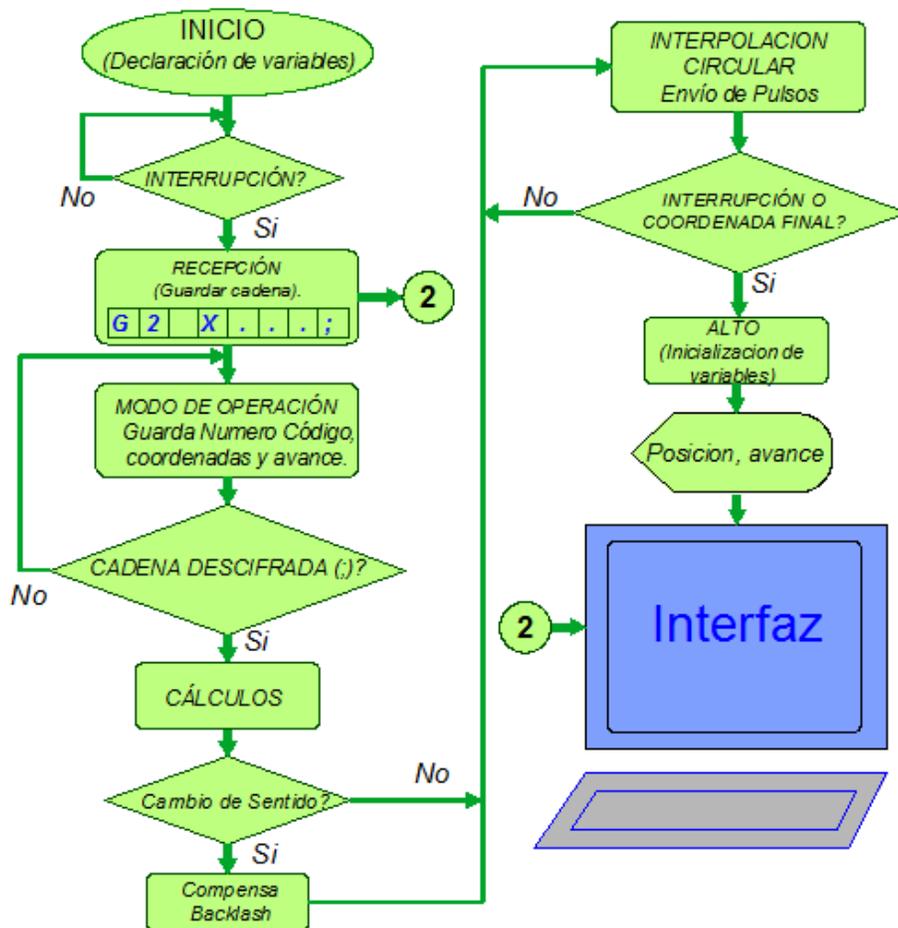


Figura 4.25 Proceso de control para un código G02.

En este ejemplo la cadena a manejar será un código *G* correspondiente a una interpolación circular en sentido horario. La ruta de operación es similar que en el ejemplo 1, pero con la diferencia que en la rutina *MODO DE OPERACIÓN*, se almacenarán más parámetros referentes a las coordenadas X, Y, I, J y F, en este caso se ejecutará la subrutina de *Interpolación circular* correspondiente al *algoritmo por búsqueda directa* descrito anteriormente. En cada iteración de este algoritmo se verifica si alcanzó las coordenadas finales.



PRUEBAS Y RESULTADOS

En este capítulo se detallan las pruebas realizadas al sistema de control descrito en el capítulo anterior. Para el desarrollo de estas pruebas se emplearon los siguientes elementos: una PC con sistema operativo Windows 7 dotado de un puerto de comunicación serial, la MMH de tipo puente y la MMH correspondiente a la segunda versión del primer prototipo desarrollado en el GMM, placas de latón de $25 \times 10 \text{ mm}$ como piezas de trabajo para realización de los maquinados, como herramientas de corte se utilizó un cortador de 0.5mm diámetro y otro 0.2 mm y un husillo neumático de fresado marca DEPRAG que opera a una velocidad máxima de 47000 rpm. La presión de operación del motor neumático se puede ajustar desde 4 a 6.3 bar con lo cual es posible manipular la velocidad de giro, potencia y torque [43].

Las pruebas al sistema de control consistieron primeramente en obtener las señales generadas por la tarjeta de control, y así poder determinar la máxima frecuencia a la que se ejecutan los algoritmos de interpolación. En seguida se realizaron pruebas de desempeño del sistema micromecánico en cada uno de los ejes, esto consistió en maquinar pequeños rectángulos para producir movimientos en un eje a la vez. El propósito de estas pruebas fue para medir el juego mecánico presente en los ejes X y Y de la MMH, y tratar de compensarlo.

Una vez que se determinó el juego mecánico de los ejes de la MMH, se procedió a realizar interpolaciones donde estuvieran presentes movimientos simultáneos en este caso se maquinaron círculos concéntricos. Y por último se realizaron maquinados de contorno del logo de los pumas y del GMM.

Todas las mediciones de los diferentes maquinados obtenidos se llevaron a cabo por medio de un comparador óptico marca "NikonProfile V-16D" con resolución de $1\mu\text{m}$, el cual cuenta con un sistema de proyección de tres lentes de aumento 20x,

50x y 100x y un área de trabajo de 50mm X 50mm. Este comparador óptico es mostrado en la figura 5.1, y se encuentra en el Laboratorio de Metrología del CCADET, UNAM.

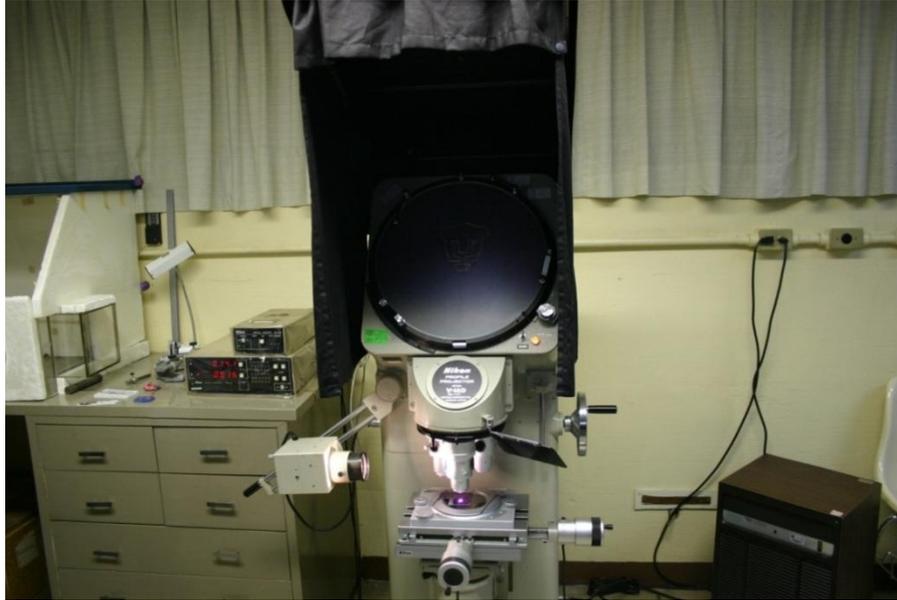


Figura 5.1. Comparador óptico del Laboratorio de Metrología (CCADET, UNAM).

5.1 Pruebas de Frecuencia de Ejecución de los Algoritmos de Interpolación.

Las pruebas realizadas a la tarjeta de control, consistieron en obtener las señales generadas durante la ejecución de los algoritmos de interpolación, con el fin de determinar la frecuencia máxima a la que se ejecutan dichos algoritmos y por consiguiente calcular las velocidades de desplazamiento de la micromáquina. Para estas pruebas se utilizaron los motores de la MMH de tipo puente y los motores de la MMH segunda versión de primer prototipo.

5.1.1 Ejecución de los Algoritmos utilizando motores de la MMH tipo puente

Los actuadores de esta MMH son motores de pasos comerciales con una resolución de 1.8° por paso. Para estas pruebas el driver de potencia R-325 se configuró para operar de la siguiente manera: suministro de corriente de 0.6 A, corriente de retención de 30%, 16 micropasos y un voltaje de carga de 30 V. Con esto se obtuvo que la frecuencia máxima a la que trabajan los actuadores con movimiento en un solo eje de 39.75 KHz, en la figura 5.2 se muestra el tren de

pulsos generado y en la figura 5.3 se ilustra los niveles de corriente en una bobina del motor trabajando en modo micropasos.

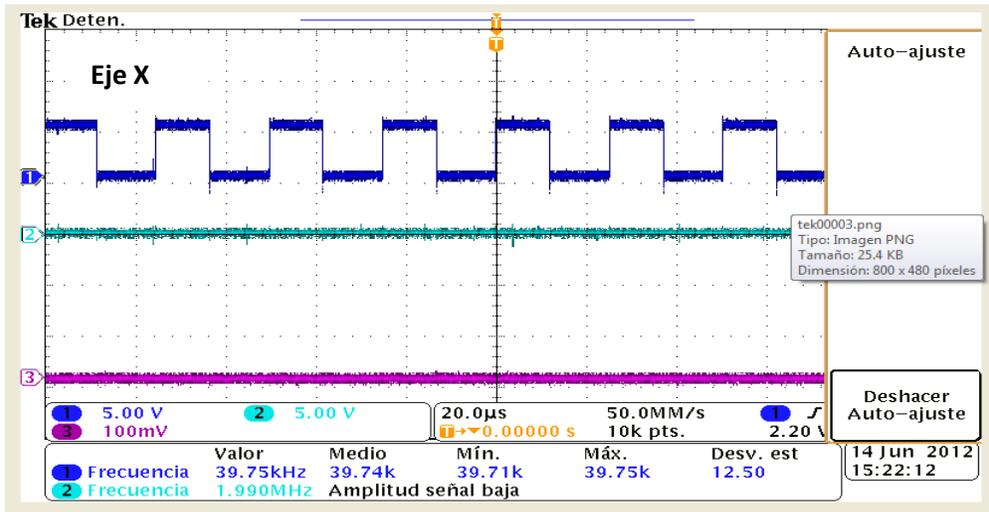


Figura 5.2. Señales generadas para una interpolación lineal en un solo eje.

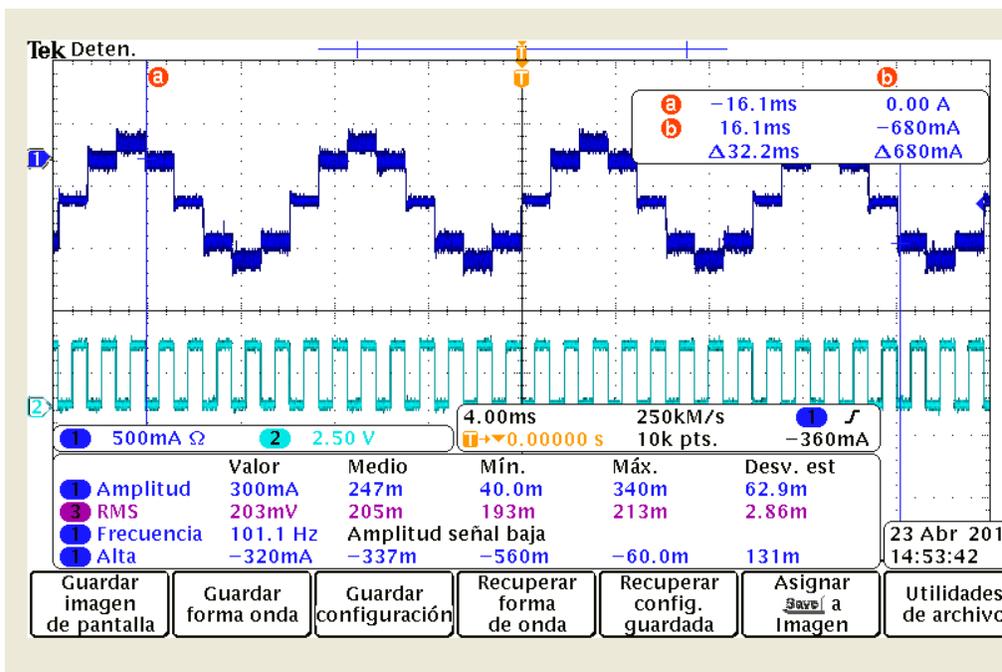


Figura 5.3. Corriente en una bobina de un motor en modo micropasos.

En la figura 5.4 se tienen registradas las señales de los ejes X y Y, al procesar un código G01 (interpolación lineal). Estas señales corresponden a la ejecución de una línea desde el origen a una posición (X=2500 μm y Y=1300 μm). Como podemos observar la velocidad del eje X es de 22 KHz y para el eje Y es de 11.5

KHz. Para este caso la velocidad en el *eje X* es aproximadamente el doble de la velocidad del *eje Y* es decir, el eje X tiene que desplazarse mucho más rápido para llegar al punto programado.

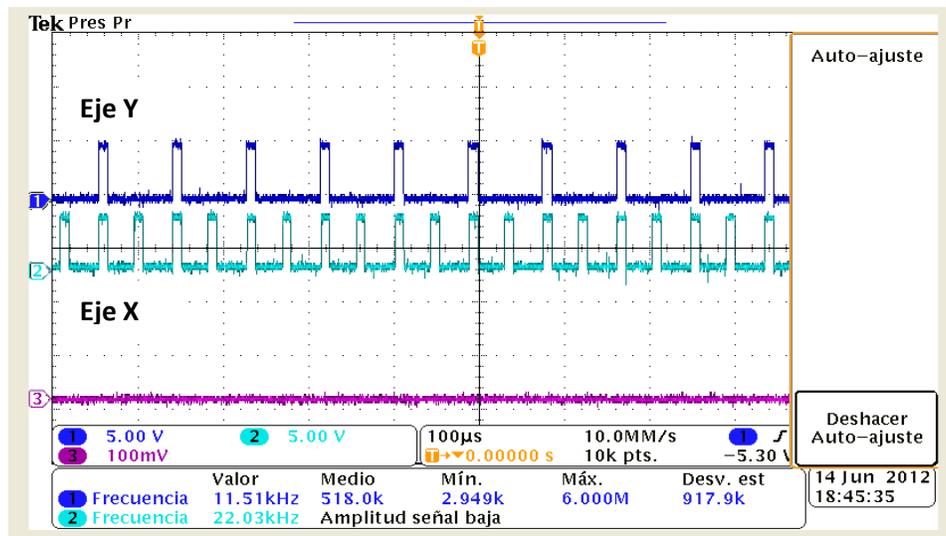


Figura 5.4 Señales generadas por un código G01.

Con la finalidad de determinar la velocidad de avance máxima a la cual se puede programar la MMH tipo puente, se programaron diferentes retardos en la ejecución de cada iteración de los algoritmos de interpolación, para cuantificar las frecuencias máximas de las señales con que se alimenta a los motores. Los resultados se ilustran en la tabla 5.1, 5.2 y 5.3.

UN SOLO EJE		
Periodo [µs]	Frecuencia(KHz)	Avance (mm/min)
10	39.75	14
11	36.82	13
12	34.29	12
13	32.09	11.2
15	28.44	10
17	25.53	9
19	23.16	8.1
22	20.34	7.1
26	17.49	6.1
32	14.46	5
38	12.32	4.3
45	10.5	3.7

Tabla 5.1. Avances para interpolación lineal en un eje para MMH tipo puente.

INTERPOLACION LINEAL XY						
Periodo (μ s) (Retardo)	Coordenada X [μ m]	Coordenada Y [μ m]	Frecuencia (KHz) Eje X	Frecuencia (KHz) Eje Y	Frecuencia Resultante (KHz)	Avance [mm/min) Resultante
9	5000	5000	32.68	32.59	46.15	16.2
10	5000	1000	32.68	6.468	33.31	11.7
11	1000	5000	6.452	32.68	33.31	11.7

Tabla 5.2 Avances para interpolación lineal dos ejes simultáneos para MMH tipo puente.

MOVIMIENTO CIRCULAR (CUARTO DE CIRCULO)				
Periodo (μ s)	Frec (KHz) Eje X	Frec (KHz) Eje Y	Frecuencia Resultante (KHz)	Avance (mm/min)
9	22.78	22.78	32.21	11.3

Tabla 5.3 Avance máximo para interpolación circular MMH tipo puente.

Para los cálculos resultantes de las tablas anteriores, se tomó la información de los parámetros mecánicos de la MMH tipo puente y parámetros de los drivers de potencia, los cuales son: 200 pasos/rev.Reducción mecánica 1:64,avance del tornillo guía 1.2, mm/rev.y16 micropasos.

A partir de esta información anterior, el tornillo guía se mueve 1.2 mm por vuelta. El motor da 64 vueltas por una vuelta del tornillo guía. El driver necesita 16 micro pasos de entrada para hacer que el motor de un pasocompleto. El driver requiere 3200 pulsos para hacer que el motor de una revolución. Por lo tanto la escala necesaria para avanzar un mm es:

$$\begin{aligned}
 \text{PulsosMM} &= \frac{200 \text{ pasos}}{1 \text{ rev. mot.}} \times \frac{16 \text{ micropasos}}{\text{paso}} \times \frac{64 \text{ rev. mot.}}{1 \text{ rev. torn.}} \times \frac{1 \text{ rev. torn.}}{1.2 \text{ mm}} \\
 &= 170667 \frac{\text{pulsos}}{\text{mm}}
 \end{aligned}$$

La resolución del sistema es el inverso de la ecuación anterior:

$$\text{Resolucion} = 5.86 \times \frac{10^{-6} \text{ mm}}{\text{pulso}} = 5.86 \text{ nm/pulso}$$

La ecuación para determinar el avance en mm/minuto es la siguiente:

$$F = \text{Resolucion} \times \text{frecuencia} \times 60$$

Dónde: La resolución está en mm/pulso y la frecuencia en pulsos/seg.

5.1.2 Ejecución de los Algoritmos utilizando motores de la 2^{da} Versión del primer prototipo de MMH.

Los motores de esta MMH fueron diseñados y fabricados en el GMM. Estos motores en la práctica operan a una frecuencia máxima de 500 Hz a paso completo. Los modos de operación para lo que fueron diseñados estos motores son a paso completo y medio paso. La resolución lineal de esta micromáquina es de 70 nm trabajando los motores a medio paso.

Para determinar la *velocidad de avance máxima* a la cual se puede programar esta MMH, se programaron diferentes retardos en la ejecución de cada iteración de los algoritmos de interpolación, para obtener señales de alimentación hacia los motores. Los resultados se ilustran en la tabla 5.4, 5.5 y 5.6.

UN SOLO EJE		
Periodo [μ s] (Retardos)	Frecuencia (Hz)	Avance (mm/min)
300	970	4
450	766	3.2
800	500	2.1

Tabla 5.4 Avance máximo en un eje para segunda versión de primer prototipo.

INTERPOLACION LINEAL XY						
Periodo (μ s) (Retardo)	Coord X [μ m]	Coord Y [μ m]	Frec (Hz) Eje X	Frec (Hz) Eje Y	Frecuencia Resultante (Hz)	Avance [mm/min] Resultante
350	5000	5000	766	766	1083	4.5
350	5000	1000	766	154	781	3.2
350	1000	5000	155	766	781	3.2

Tabla 5.5 Avance máximo en una interpolación lineal con segunda versión de primer prototipo.

MOVIMIENTO CIRCULAR (CUARTO DE CIRCULO)				
Periodo (μ s)	Frec (Hz) Eje X	Frec (Hz) Eje Y	Frecuencia Resultante (Hz)	Avance (mm/min)
350	760	760	1075	4.5

Tabla 5.6 Avance máximo en una interpolación circular con segunda versión de primer prototipo.

5.2 Maquinados con MMH tipo Puente para Determinar Juego Mecánico

El prototipo de la figura 5.5 tiene una configuración de microcentro de maquinado vertical tipo puente móvil, cuenta con dimensiones exteriores de $110 \times 110 \times 100$ mm, posee 3 ejes ortogonales traslacionales y un volumen de trabajo de $20 \times 20 \times 20$ mm. Los actuadores de esta MMH son motores de pasos comerciales con una resolución de 1.8° por paso, la reducción del sistema de transmisión es de 1:64 y la corriente máxima de operación de los motores es de 0.6 A.

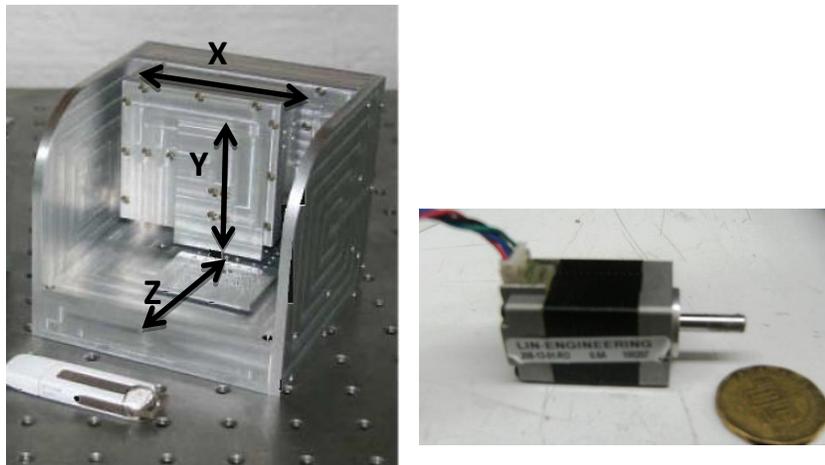


Figura 5.5 MMH tipo puente y motor de pasos empleado [42].

De acuerdo a las pruebas de juego mecánico realizadas a este prototipo en cada uno de los ejes, se reporta en el trabajo de tesis *“Investigación y Evaluación de Mecanismos Para el Desarrollo de Micromáquinas Herramienta”* [42], que el error lineal del sistema, referente al juego mecánico es de $2.24 \mu\text{m}$. Dichas pruebas se llevaron a cabo sin carga en los motores. Con esto se espera que los maquinados a realizarse con este prototipo tengan un error dimensional de entre 2 y $3 \mu\text{m}$.

Los maquinados llevados a cabo con la MMH tipo puente utilizando un cortador de 0.5mm de diámetro, 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 realizaron maquinados en dos regiones verticales (V1, V2) y dos horizontales (H1, H2). Las dimensiones del área de trabajo son: 15mm en el eje X y 7mm en el eje Y.

Las mediciones se realizaron tomando la diferencia de longitudes entre el punto de inicio y punto final, y las unidades son en μm . Los resultados de los maquinados se muestran en la figura 5.6 y las mediciones obtenidas se registran en la tabla 5.7.

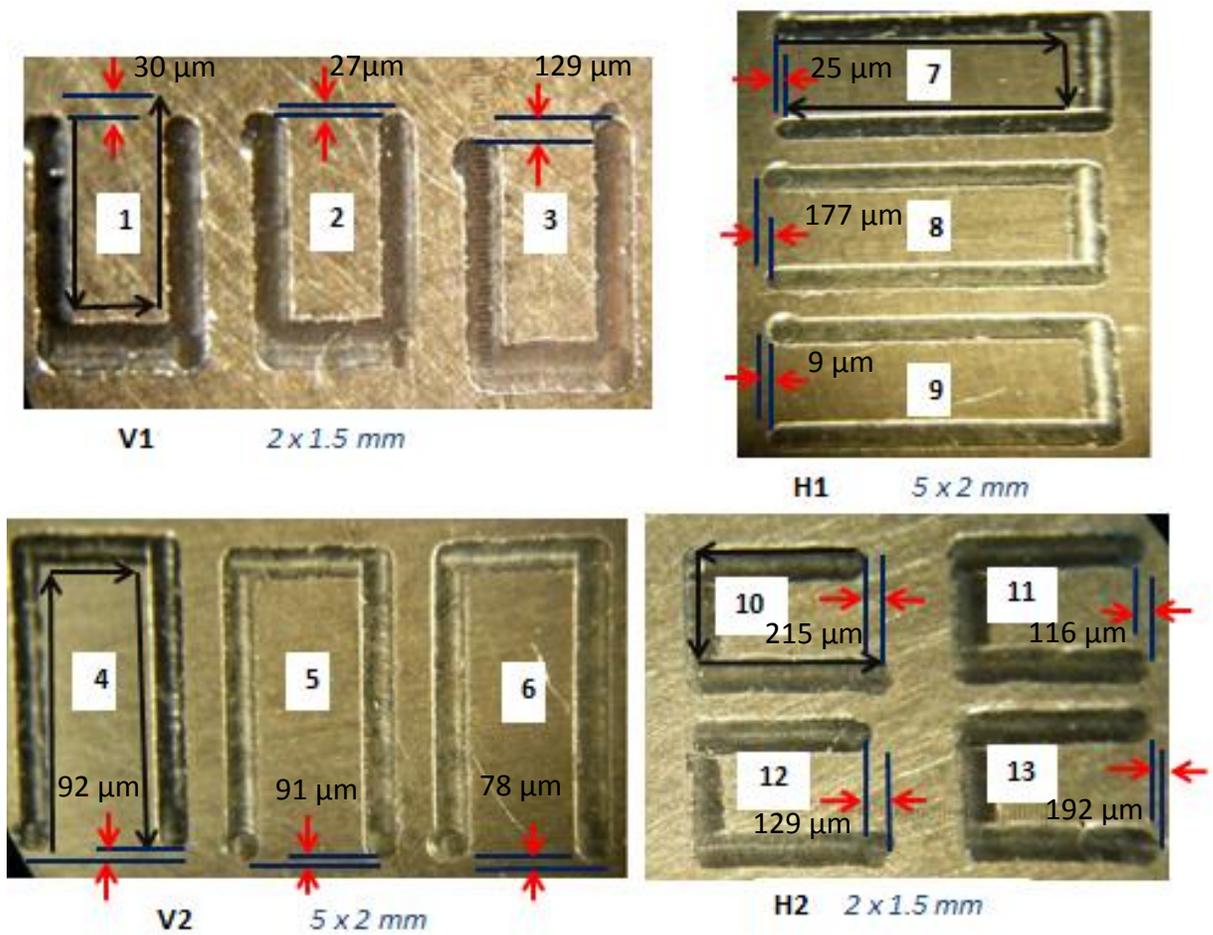


Figura 5.6 Resultado de Maquinados con MMH tipo puente.

Figura		Juego Mecánico [μm]
1	Eje Y positivo	30
2	Eje Y positivo	-27
3	Eje Y positivo	-129
4	Eje Y negativo	92
5	Eje Y negativo	91
6	Eje Y negativo	78
7	Eje X negativo	25
8	Eje X negativo	177
9	Eje X negativo	9
10	Eje X positivo	-215
11	Eje X positivo	116
12	Eje X positivo	-129
13	Eje X positivo	-192

Tabla 5.7 Magnitudes de Juego mecánico en los ejes X y Y con MMH tipo puente.

De acuerdo a las mediciones realizadas en los maquinados, se observa que el maquinado de la figura 2 y 3, la trayectoria vertical donde inicia el maquinado es mayor $27\ \mu\text{m}$ y $129\ \mu\text{m}$ respectivamente, que la trayectoria vertical donde termina, el signo negativo indica que hubo pérdida de desplazamiento durante el trayecto en la dirección negativa, que no debería de ocurrir, por lo que implica que existió una pérdida de pasos en el sistema de transmisión de la máquina, lo cual *no pertenece al juego mecánico* entre engranes. Este mismo caso se presentó en la zona H2 en las figuras 10, 12 y 13.

Debido a la presencia de este fenómeno no fue posible determinar realmente el juego mecánico. Se concluyó que las pérdidas de pasos se deben al tipo de material con que está construido el sistema de transmisión de la máquina, el cual es de plástico, y que podría sufrir deformaciones y tener cierto grado de desgaste en el momento de realizar las pruebas con carga. Ante esta situación se decidió utilizar la segunda versión del primer prototipo de MMH.

5.3 Maquinados con 2^{da} versión del primer prototipo de MMH

En la figura 5.7, se muestra la segunda versión del primer prototipo de MMH.

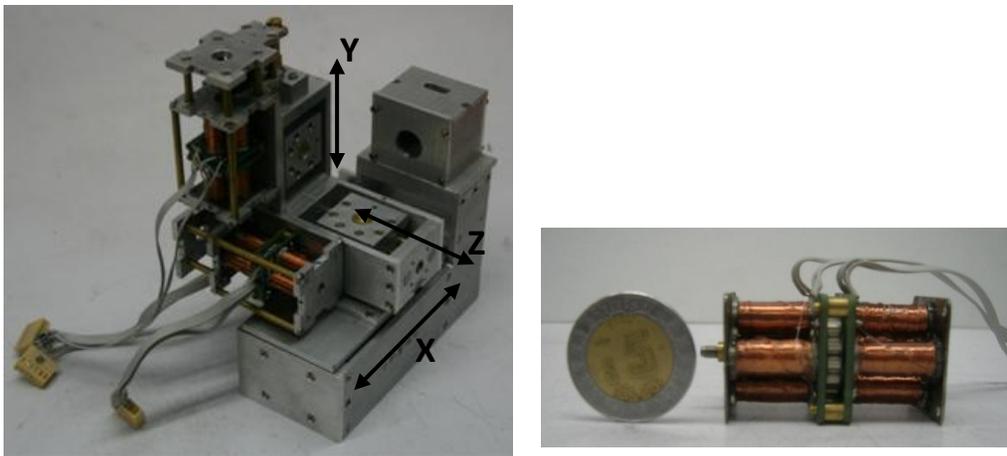


Figura 5.7 Segunda versión del primer prototipo de MMH y motor de pasos empleado [24].

Este prototipo tiene dimensiones de $120 \times 160 \times 85\ \text{mm}$, tres ejes ortogonales traslacionales, y uno rotacional. La relación de transmisión de engranes es de 1:128, sus actuadores son motores de pasos con resolución angular 18° por paso, equivalente a 20 pasos por revolución. La resolución lineal de este prototipo es de $140\ \text{nm}$ por paso del motor. La corriente máxima de operación de los motores es de 0.35 A.

Con anterioridad este prototipo se sometió a una caracterización con el fin de determinar el juego mecánico y su respectiva resolución en cada uno de los ejes. Las pruebas se realizaron operando los motores sin carga. La resolución obtenida fue de 140 nm por paso del motor operando a paso completo, y el juego mecánico en cada uno de los ejes tanto en el sentido positivo como negativo fueron los siguientes:

$$\text{JuegoXpositivo} = 275.13\ \mu\text{m}$$

$$\text{JuegoXnegativo} = 266.34\ \mu\text{m}$$

$$\text{JuegoYpositivo} = 245.19\ \mu\text{m}$$

$$\text{JuegoYnegativo} = 243.77\ \mu\text{m}$$

$$\text{JuegoZpositivo} = 201\ \mu\text{m}$$

$$\text{JuegoZnegativo} = 201.6\ \mu\text{m}$$

Los errores presentes en cada uno de los ejes realizando la compensación del juego mecánico en cada uno de los ejes fueron los siguientes:

Error Eje X Positivo entre $-6\ \mu\text{m}$ a $6\ \mu\text{m}$.

Error Eje X negativo entre $-6\ \mu\text{m}$ a $6.5\ \mu\text{m}$.

Error Eje Y Positivo entre $-8\ \mu\text{m}$ a $10\ \mu\text{m}$.

Error Eje Y negativo entre $-9\ \mu\text{m}$ a $9\ \mu\text{m}$.

En las figuras 5.8 y 5.9 se ilustra gráficamente el error en los ejes X y Y.

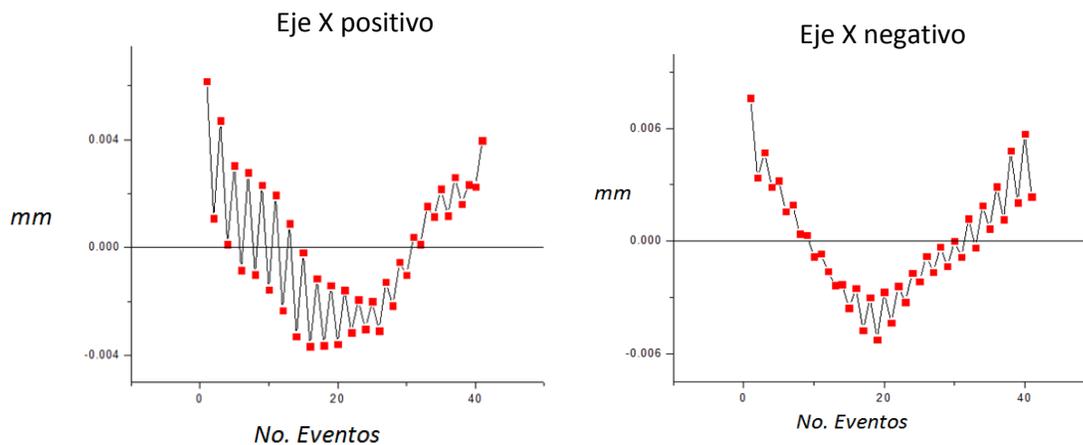


Figura 5.8 Errores presentes en el eje X

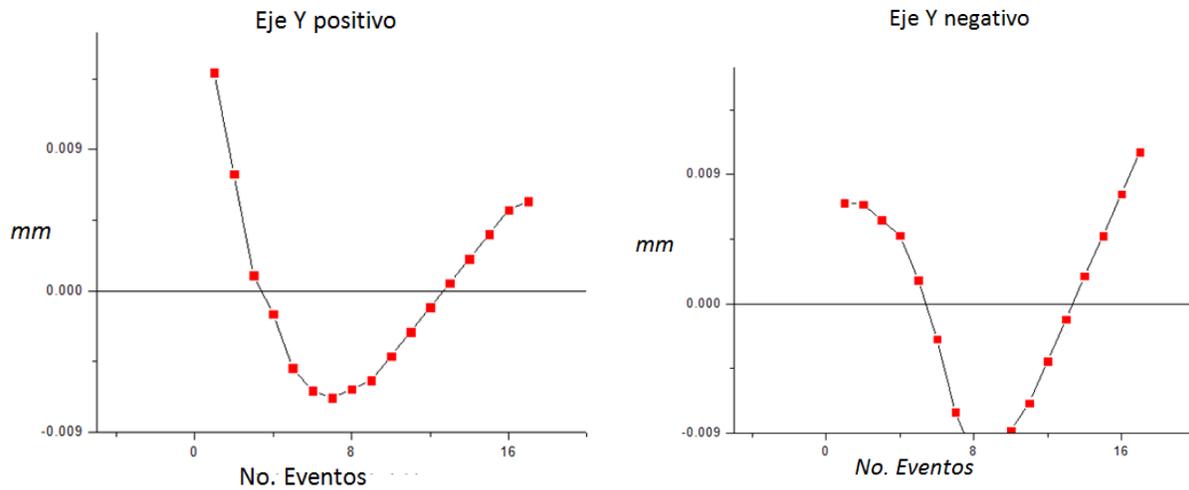


Figura 5.9 Errores presentes en el eje Y.

Los maquinados que se realizaron con esta MMH consistieron en dos rectas paralelas, unidas en uno de sus extremos por otra recta, para cada uno de los ejes en cuestión X y Y. Estos maquinados se efectuaron compensando el juego mecánico existente, con la finalidad de corroborar los valores de compensación. La zona donde se efectuaron los maquinados fue en la parte central del área de trabajo de la micromáquina (10 mm x 7 mm). Las dimensiones de los rectángulos son de 2.2 x 1.5mm y se utilizó un cortador de 0.5 mm de diámetro.

La ejecución de este proceso consistió en comenzar el maquinado en un sentido y terminar con movimiento en dirección opuesta, y posteriormente realizar el mismo proceso pero a la inversa, esto en cada uno de los ejes.

A continuación presentamos los diagramas de los maquinados, así como también las fotografías correspondientes. Con estos resultados se determinó que existe una diferencia de longitudes en las rectas paralelas correspondiente al juego mecánico. Con base a estos valores se obtuvo la media aritmética a través de la siguiente ecuación:

$$\bar{X} = \frac{1}{N} \sum_{i=1}^N x_i \quad (5.1)$$

En la figuras 5.10 se ilustran las figuras de los maquinados en el eje Y, comenzando en sentido negativo y finalizando en sentido positivo, y en la tabla 5.8 se enlistan los valores correspondientes al juego mecánico así como también el promedio de estos.

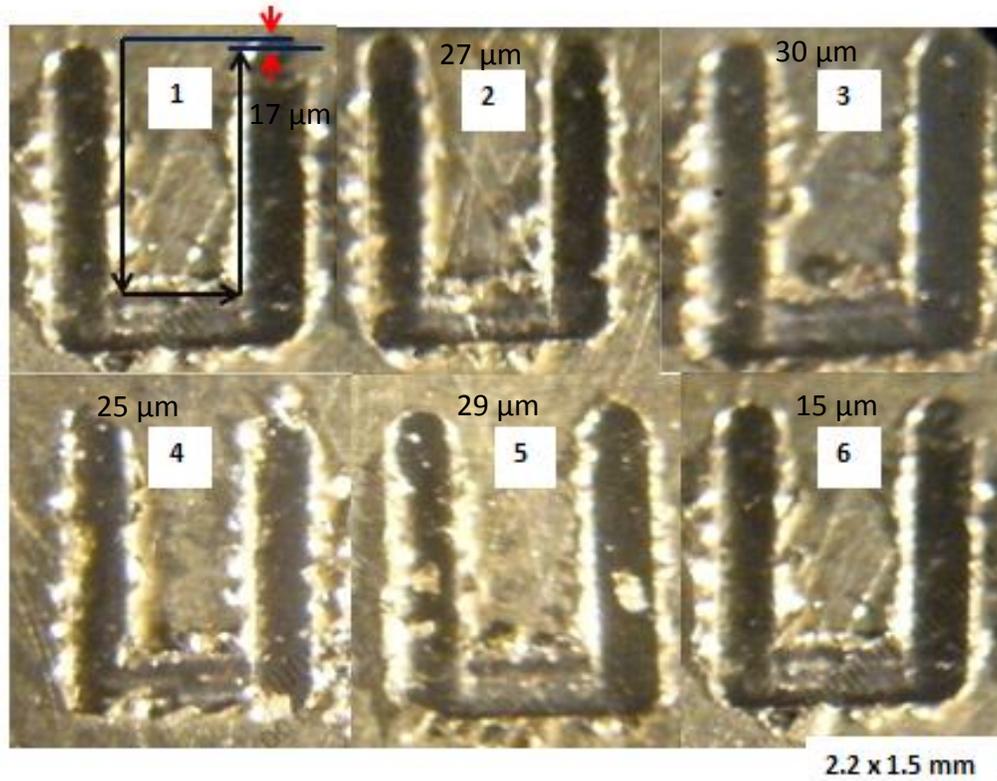


Figura 5.10 Maquinados para determinar juego mecánico en el eje Y positivo.

Figura	Juego Mecánico Eje Y positivo [μm]	Media aritmética [μm]
1	17	23.8
2	27	
3	30	
4	25	
5	29	
6	15	

Tabla 5.8 Magnitudes de Juego mecánico para el eje Y positivo.

Ahora en las figuras 5.11 se muestran los maquinados correspondientes al juego mecánico en el eje Y negativo, y en la tabla 5.9 se enlistan las magnitudes y el promedio.

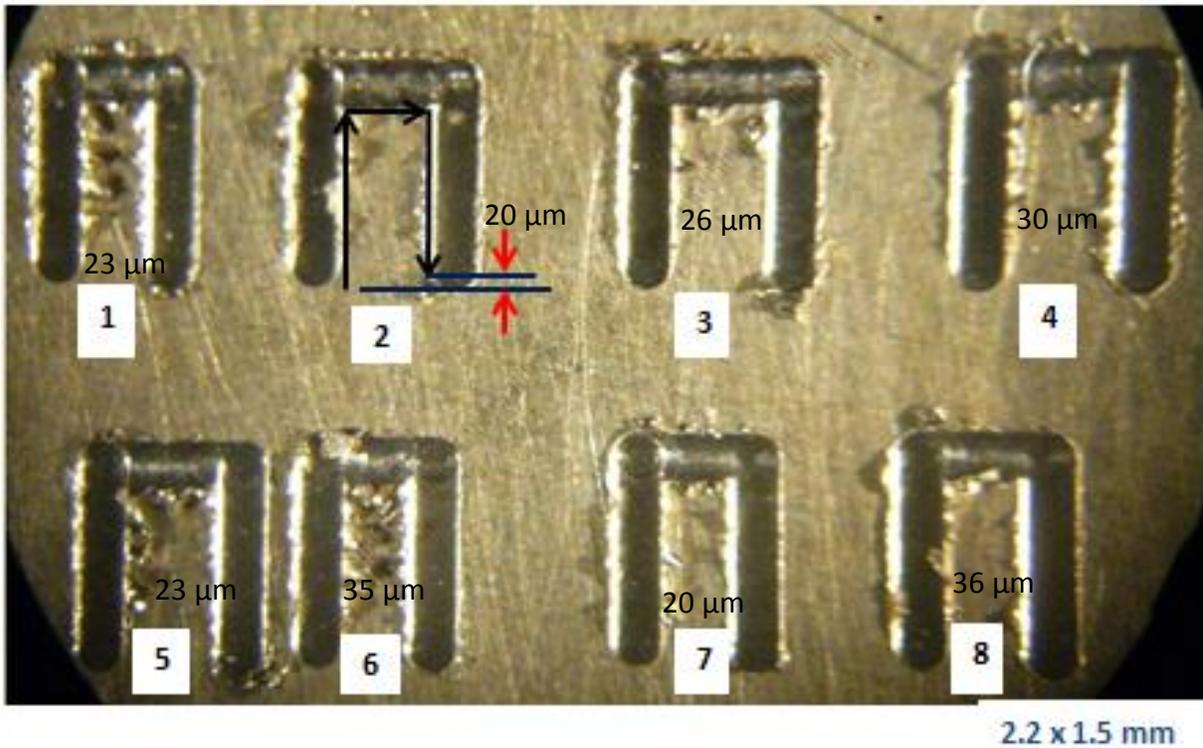


Figura 5.11 Maquinados para determinar juego mecánico en el eje Y negativo.

Figura	Juego Mecánico Eje Y negativo [μm]	Media aritmética [μm]
1	23	26.6
2	20	
3	26	
4	30	
5	23	
6	35	
7	20	
8	36	

Tabla 5.9 Magnitudes del Juego mecánico en el eje Y negativo.

En las figura 5.12 se muestran los maquinados correspondiente al juego mecánico en el eje X negativo, y en la tabla 5.10 se describen las magnitudes y el promedio.

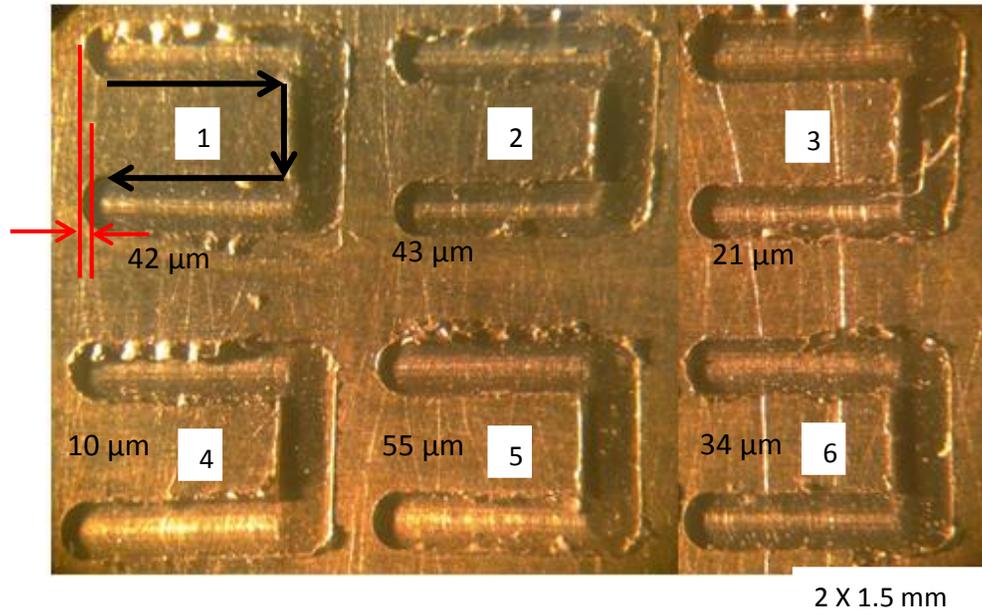


Figura 5.12 Maquinados para determinar juego mecánico en el eje X negativo.

Figura	Juego Mecánico Eje X negativo [μm]	Media aritmética [μm]
1	42	34.16
2	43	
3	21	
4	10	
5	55	
6	34	

Tabla 5.10 Magnitudes del Juego mecánico en el eje X negativo.

Por último en la figura 5.13 se ilustran los maquinados correspondientes al juego mecánico en el eje X positivo y en la tabla 5.11 los valores correspondientes.

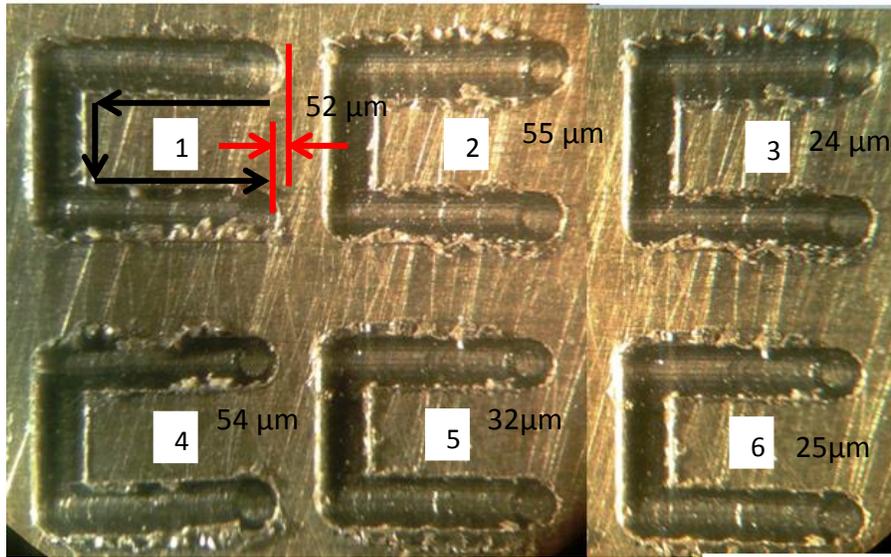
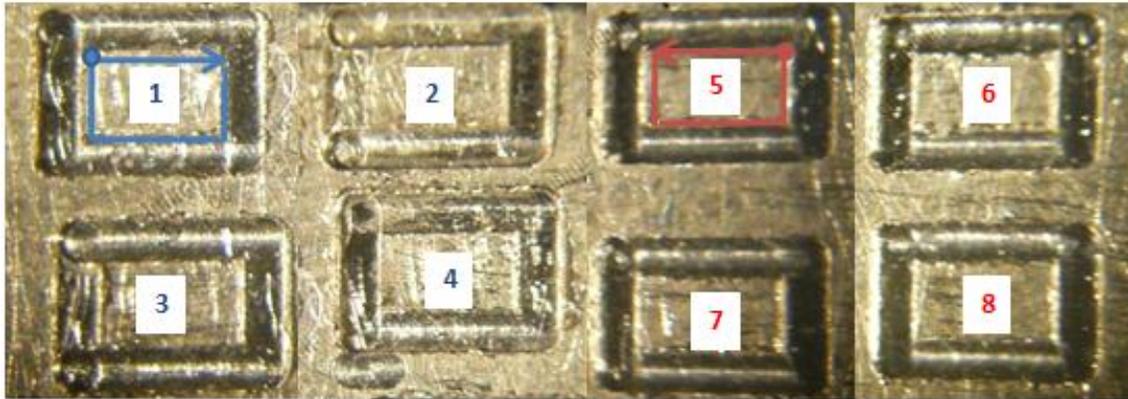


Figura 5.13 Maquinados para determinar juego mecánico en $2 \times 1.5 \text{ mm}$

Figura	Juego Mecánico Eje X positivo [μm]	Media aritmética [μm]
1	52	40.3
2	55	
3	24	
4	54	
5	32	
6	25	

Tabla 5.11 Magnitudes del Juego mecánico en el eje X positivo.

Con el fin de corroborar los valores de compensación del juego mecánico, se procedió a maquinar ocho cuadrados de dimensiones de $2.2 \times 1.5 \text{ mm}$ en la zona central del área de trabajo. El maquinado de estos cuadrados se llevó a cabo realizando 2 pasadas con una profundidad de $40 \mu\text{m}$ cada una. En la figura 5.14 se muestran dichos cuadrados y en la tabla 5.12 las dimensiones experimentales obtenidas.



2.2 x1.5 mm

Figura 5.14 Maquinados para comprobar el juego mecánico.

Figura		Dimensiones teóricas[mm]	Dimensiones experimentales [mm]	Juego mecánico [μm]	Promedio [μm]
1	Eje X-	2.200	2.190	10	Eje X negativo 3.75
	Eje Y+	1.500	1.187	13	
2	Eje X-	2.200	2.203	-3	
	Eje Y+	1.500	1.504	-4	
3	Eje X-	2.200	2.201	-1	Eje Y positivo 4.5
	Eje Y+	1.500	1.505	-5	
4	Eje X-	2.200	2.196	9	
	Eje Y+	1.500	1.186	14	
5	Eje X+	2.200	2.197	3	Eje X positivo 4.25
	Eje Y-	1.500	1.498	2	
6	Eje X+	2.200	2.200	0	
	Eje Y-	1.500	1.506	-6	
7	Eje X+	2.200	2.195	5	Eje Y negativo 3.5
	Eje Y-	1.500	1.490	10	
8	Eje X+	2.200	2.191	9	
	Eje Y-	1.500	1.492	8	

Tabla 5.12 Resultados de los maquinados de la figura 5.15.

Como se observa, la compensación realizada aún no compensa totalmente el juego mecánico, de tal forma que obtuvimos nuevamente los promedios. Los promedios resultantes se le suman a los ya existentes obteniendo nuevos valores de compensación. En consecuencia los valores finales de juego mecánico son:

$$\text{JuegoXpositivo} = 275.13 + 34.6 + 4.2 = 314 \mu\text{m}$$

$$\text{JuegoXnegativo} = 266.34 + 34.16 + 4.4 = 305.3 \mu\text{m}$$

$$\text{JuegoYpositivo} = 245.19 + 23.8 - 4.5 \mu\text{m} = 264.5 \mu\text{m}$$

$$\text{JuegoYnegativo} = 243.77 + 26.6 + 3.5 \mu\text{m} = 273.8 \mu\text{m}$$

Con estos nuevos valores de juego mecánico se procedió a maquinar rectas paralelas de 2mm de longitud y separación de 40μm con movimiento de zigzag, utilizando un cortador de 0.5 mm de diámetro. En la figura 5.15 se muestran dichos maquinados y en la tabla 5.13 las mediciones experimentales registradas.

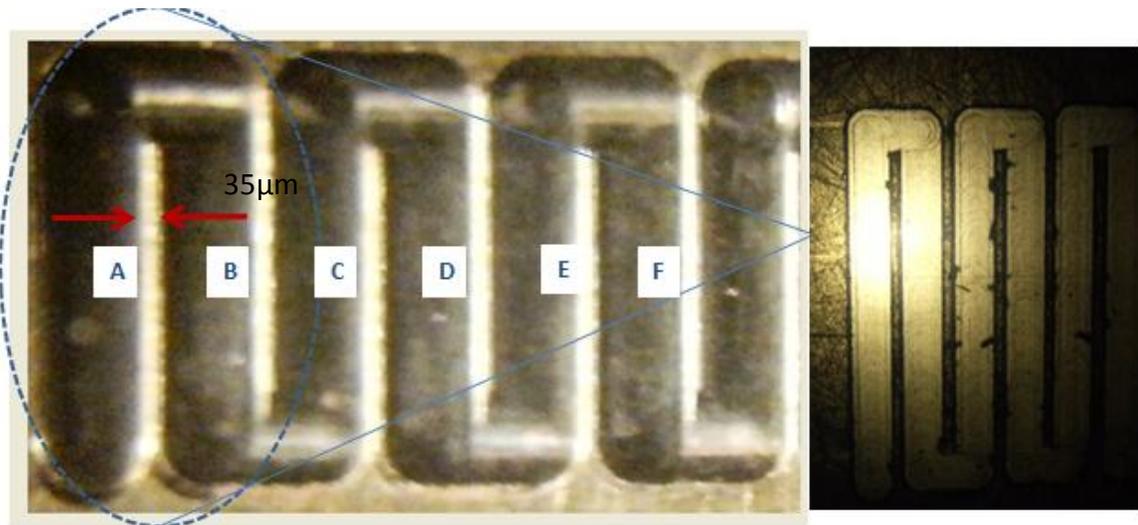


Figura 5.15 Maquinados con movimiento de zigzag con cortador de 0.5 mm.

Distancia	Dimensión teórica [μm]	Dimensión experimental [μm]	Error [μm]
A	40	35	5
B	40	36	4
C	40	38	2
D	40	37	3
E	40	39	1
F	40	36	4

Tabla 5.13 Resultados de los maquinados de la figura 5.11.

Como se puede observar en los resultados de la tabla 5.13, el error máximo es de 5 μm, con esto se demuestra que este error está dentro de los valores obtenidos en la caracterización de la MMH.

Posteriormentese maquinaron rectas paralelas de 2 mm de longitud y con una separación de 70 μm utilizando un cortador de 0.2 mm de diámetro, con movimientos de zigzag tanto en el eje X y Y. Los resultados de estos maquinados se presentan en las figura 5.16 y 5.17, y las mediciones de los resultados experimentales se describen en las tablas 5.14 y 5.15 respectivamente.

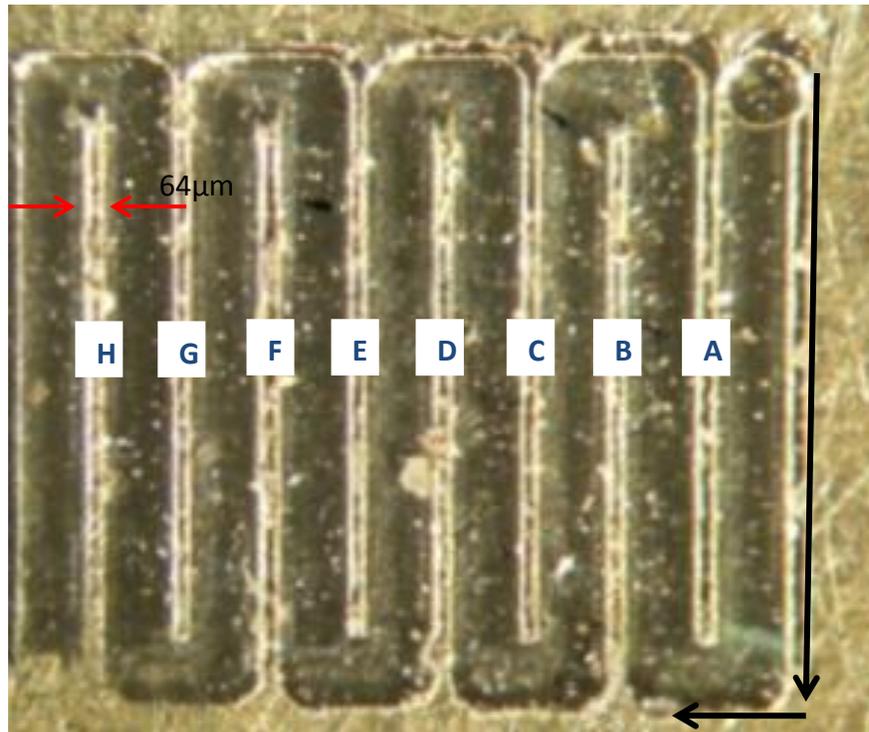


Figura 5.16 Maquinados con movimiento de zigzag en eje X.

Distancia	Dimensión teórica [μm]	Dimensión experimental [μm]	Error [μm]
A	70	64	6
B	70	62	8
C	70	71	1
D	70	63	7
E	70	64	6
F	70	68	2
G	70	66	4
H	70	67	3

Tabla 5.14 Resultados de los maquinados de la figura 5.12.

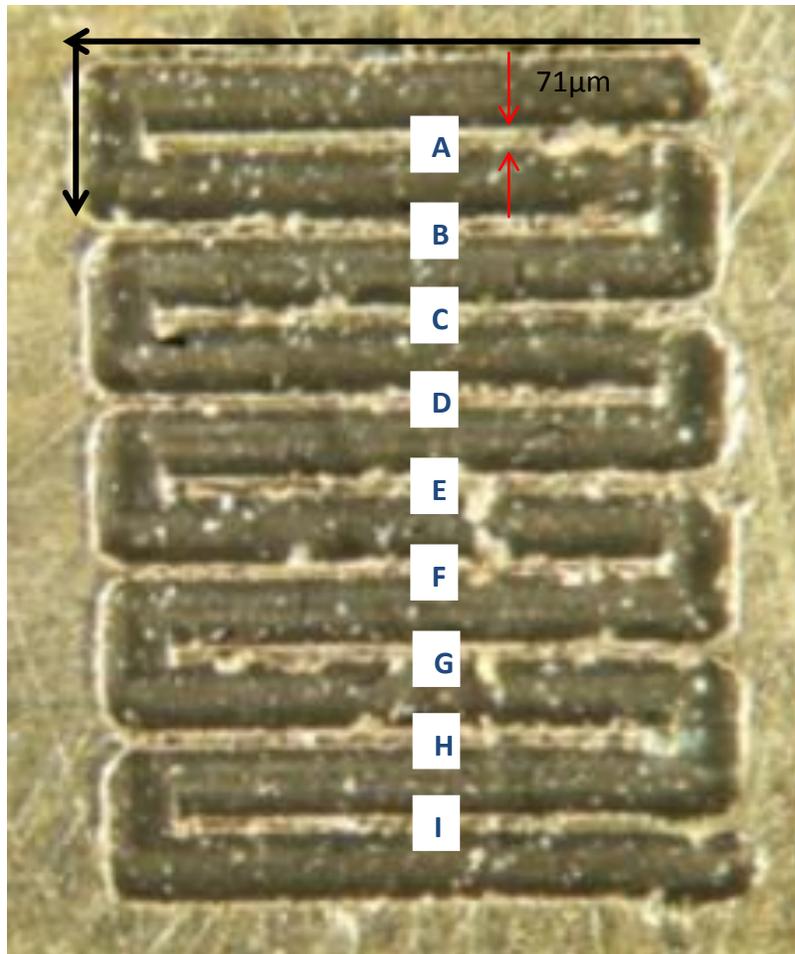


Figura 5.17 Maquinados con movimiento de zigzag en eje Y.

Distancia	Dimensión teórica [μm]	Dimensión experimental [μm]	Error [μm]
A	70	71	1
B	70	63	7
C	70	75	5
D	70	62	8
E	70	72	2
F	70	76	6
G	70	71	1
H	70	63	7
I	70	66	4

Tabla 5.14 Resultados de los maquinados de la figura 5.12.

Se observa que los maquinados de la figuras 5.16 presentan un error máximo de 8 μm cuando el movimiento se realiza en el eje X, y para los maquinados de la figura 5.17 también el error máximo obtenido es de 8 μm .

Posteriormente con el fin de comprobar los algoritmos de interpolación se prosiguió a maquinar figuras que requieren de movimientos simultáneos en dos ejes.

5.4 Maquinados con interpolación en dos ejes simultáneos.

Con la finalidad cuantificar el nivel de aproximación de los algoritmos de control programados en la tarjeta de control, las geometrías propuestas a maquinar consistieron en 3 círculos concéntricos (1), (2),(3)de 1, 2, y 3 mm de diámetro respectivamente. El objetivo es verificar la capacidad del sistema propuesto para producir de manera repetitiva un determinado diámetro. Los maquinados resultantes (realizados en sentido horario) se muestran en la figura 5.18, y las magnitudes de los diámetros (\varnothing_1 , \varnothing_2 , \varnothing_3) correspondientes a cada circunferencia son registradas en la tabla 5.15.

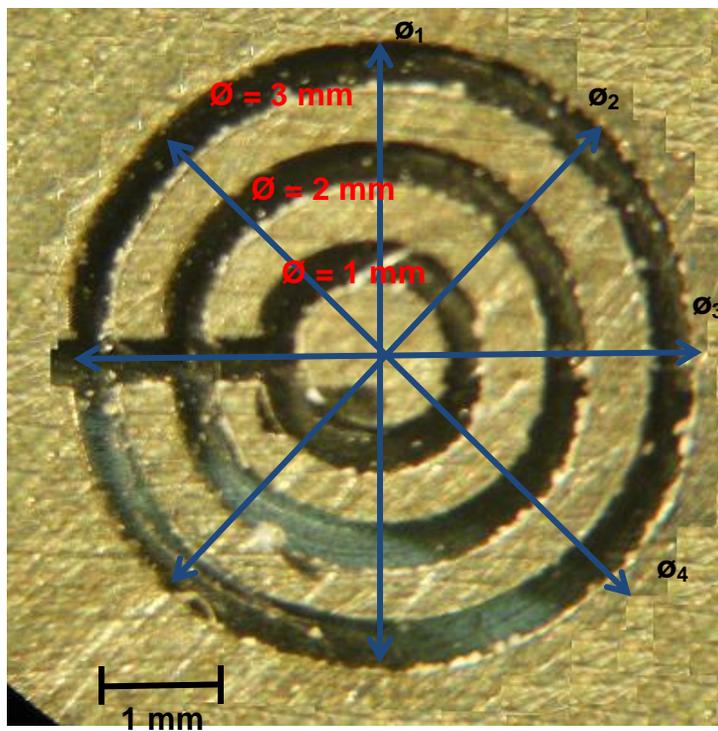


Figura 5.18 Maquinado de tres círculos concéntricos

También en la tabla se muestra la media aritmética y la desviación estándar de las mediciones de cada círculo, la desviación estándar, σ , es una medida de dispersión que nos indica que tan alejadas se encuentran las mediciones con respecto a la media aritmética, y se calcula con la siguiente expresión:

$$\sigma = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})^2} \quad (5.2)$$

Otro concepto estadístico que tiene utilidad para cuantificar la dispersión de los datos es el *coeficiente de variación*, *c.v.* el cual se define como el cociente de la desviación estándar entre la media. De esta manera proporciona una medición normalizada de la dispersión, con frecuencia se multiplica por 100 para expresarlo como porcentaje. Este coeficiente de determina con la siguiente expresión:

$$c. v = \frac{\sigma}{\bar{x}} \times 100 \quad (5.3)$$

Figura	Diámetro teórico [mm]	Diámetro experimental [mm]	Promedio [mm]	Desviación estándar [mm]	Coficiente de variación [%]
1	1	$\varnothing_1=0.955$ $\varnothing_2=0.941$ $\varnothing_3=0.938$ $\varnothing_4=0.991$	0.956	0.024	2.5
2	2	$\varnothing_1=1.949$ $\varnothing_2=1.966$ $\varnothing_3=1.909$ $\varnothing_4=1.982$	1.951	0.031	1.5
3	3	$\varnothing_1=2.929$ $\varnothing_2=2.965$ $\varnothing_3=2.904$ $\varnothing_4=2.978$	2.944	0.033	1.1

Tabla 5.15 Resultados de los maquinados de la figura 5.14.

De acuerdo a los resultados obtenidos en la tabla 5.15, podemos deducir que los errores presentes, referente a la repetibilidad en los maquinados de una circunferencia, para el círculo de diámetro de 1 mm existe un error del 2.5%, mientras que para los círculos de 2 y 3 mm de diámetro prevalece un error de 1.5 y 1.1% respectivamente.

Posteriormente se realizó un segundo experimento realizando maquinados con circunferencias de diámetros 1, 2, 3, 4 y 5 mm, con el objetivo de obtener una estimación más precisa de las condiciones de operación de la máquina cuando opera con dos ejes simultáneamente. En la figura 5.19 se ilustran dichos maquinados y en la tabla 5.16 se describen las mediciones obtenidas.

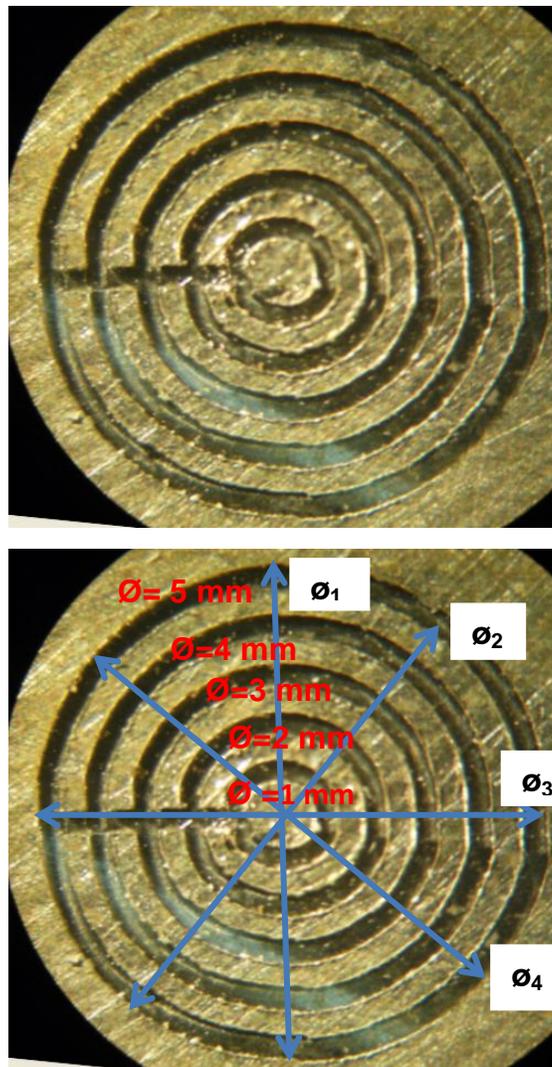


Figura 5.19 Maquinado de 5 círculos concéntricos.

Figura	Diámetro teórico [mm]	Diámetro experimental [mm]	Promedio [mm]	Desviación estándar [mm]	Coefficiente de variación [%]
1	1	$\varnothing_1=0.940$ $\varnothing_2=0.929$ $\varnothing_3=0.916$ $\varnothing_4=0.988$	0.943	0.031	3.2
2	2	$\varnothing_1=1.925$ $\varnothing_2=1.930$ $\varnothing_3=1.907$ $\varnothing_4=1.986$	1.937	0.034	1.7
3	3	$\varnothing_1=2.917$ $\varnothing_2=2.924$ $\varnothing_3=2.897$ $\varnothing_4=2.974$	2.928	0.034	1.1
4	4	$\varnothing_1=3.904$ $\varnothing_2=3.952$ $\varnothing_3=3.893$ $\varnothing_4=3.967$	3.929	0.036	0.9
5	5	$\varnothing_1=4.883$ $\varnothing_2=4.928$ $\varnothing_3=4.877$ $\varnothing_4=4.945$	4.908	0.033	0.6

Tabla 5.16 Resultados de los maquinados de la figura 5.16.

Haciendo una comparación de la tablas 5.15 y 5.16, se observa que el error en la circunferencia 1 incrementó de 2.5% a 3.2%, esto se puede descifrar visualmente en la figura 5.16 la deformación presente en dicha circunferencia, mientras que el error en los demás círculos permanece igual, con esto podemos concluir que existe una buena repetibilidad en los maquinados.

5.5 Procedimiento para la Ejecución de un programa CNC

El procedimiento para la manufactura de las geometrías propuestas utilizando la interfaz gráfica y el sistema de control desarrollado fue el siguiente:

1. Dibujar los contornos de la figura en un programa de *Diseño Asistido por Computadora*.
2. Importar el dibujo en un software CAD-CAM y generar el proceso de manufactura.
3. Generar el código G y M.
4. Depurar de forma manual el código obtenido y guardar el archivo con extensión .txt.
5. Abrir desde la interfaz gráfica el archivo guardado. (El programa aparece en el cuadro denominado PROGRAMA CNC).
6. Ejecutar la simulación para verificar el proceso de manufactura.
7. Determinar el cero de pieza.
8. Ejecutar programa CNC.

5.5.1 Diseño en CAD

El software empleado para realizar el dibujo del logo de los pumas y el logo del GMM fue AutoCAD 2012, ver figura 5.20. Las dimensiones de esta geometría es 4.73 mm en el eje X y 4.83 mm en el eje Y.

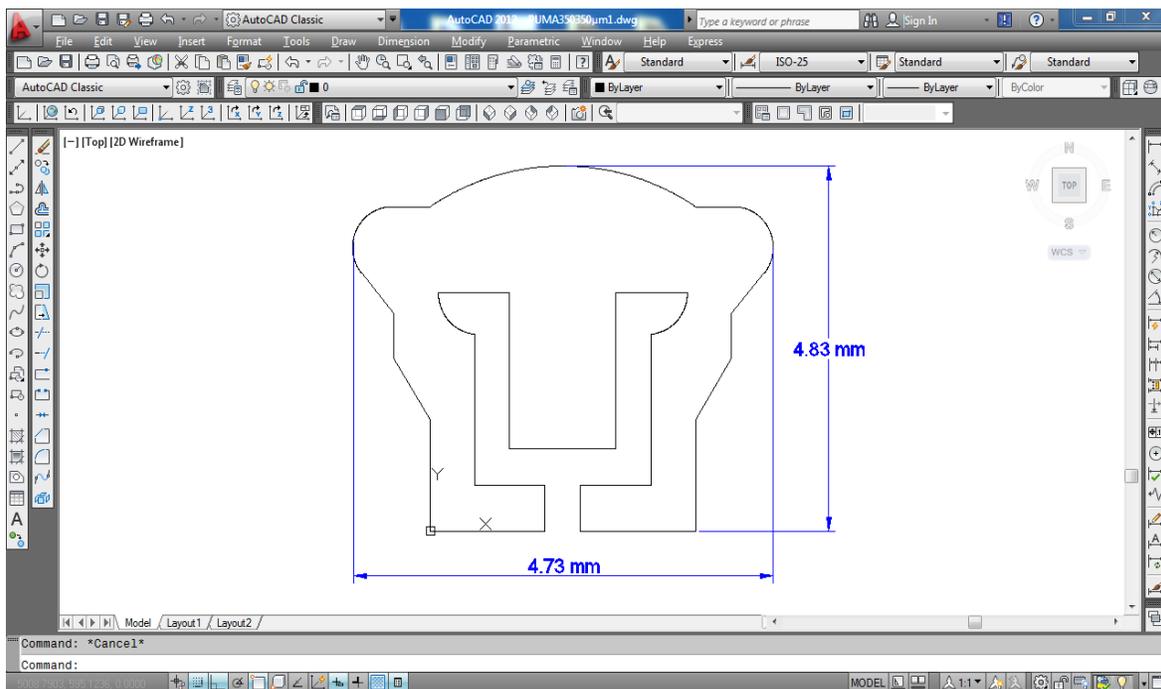


Figura 5.20 Escudo de los Pumas hecho en AutoCAD.

5.5.2 Proceso de Manufactura

Una vez que se ha concluido el diseño de la pieza, mediante un software CAD-CAM se importa el dibujo para generar el proceso de manufactura. El software

CAD-CAM empleado para generar el código referente al proceso de manufactura de la figura correspondiente, fue Mastercam X5. El proceso de maquinado corresponde a una *operación de contorneado* utilizando como herramienta un cortador de 0.2 mm de diámetro. En la figura 5.21 se muestra la pantalla del ambiente de Mastercam y se observa la simulación del recorrido físico de la herramienta.

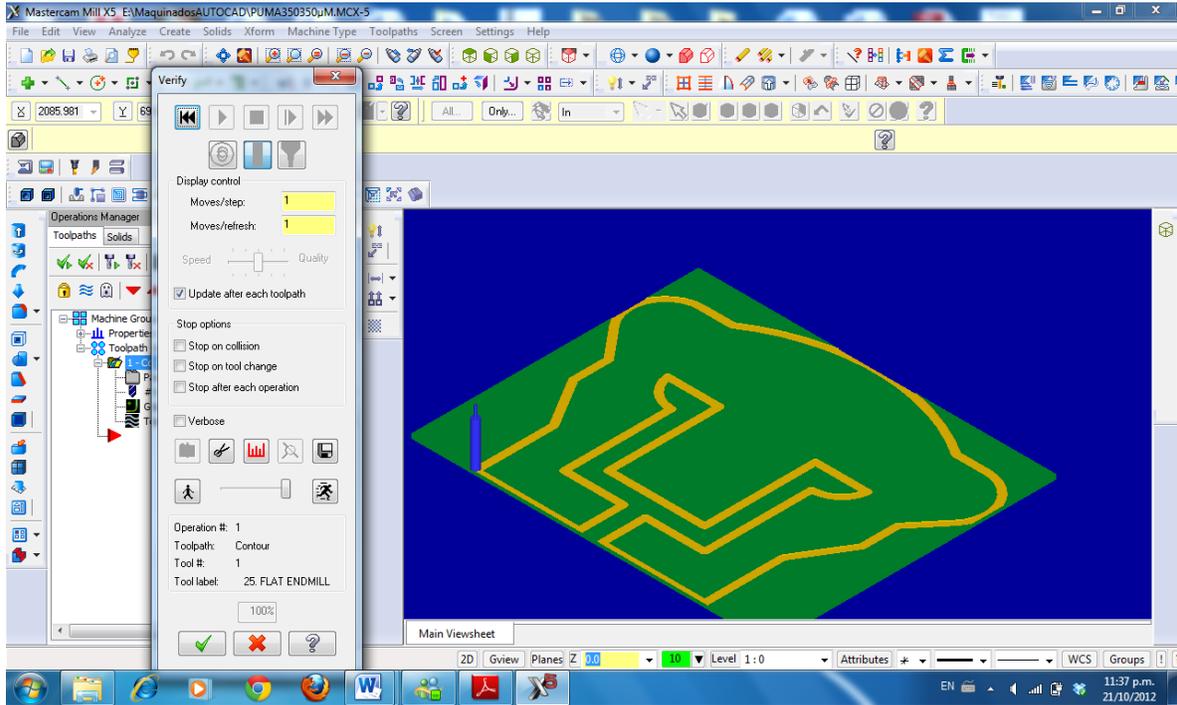


Figura 5.21 Trayectoria del proceso de manufactura del Puma en Mastercam.

Una vez que se ha definido la trayectoria del proceso de manufactura se puede obtener el programa para maquinado utilizando un postprocesador adecuado, en la figura 5.22 se observa el código generado el cual puede guardarse con extensión *.NC o *.txt.

```
(NC FILE - C:\USERS\SIMONMARTINEZC\DOCUMENTS\MY MCAMX5\MILL\NC\PUMA350350µM.NC)
(MATERIAL - ALUMINUM MM - 2024)
( T1 | 25. FLAT ENDMILL | H0 )
N0 G21
N1 G0 G17 G40 G49 G80 G90
N2 T1 M6
N3 G0 G90 G54 X-50. Y-50. A0. S1527 M3
N4 G43 H0 Z25.
N5 Z10.
N6 G1 Z-10. F305.4
N7 X1350. F610.8
N8 Y550.
N9 X550.
N10 Y1997.482
N11 X505.069 Y2002.061
N12 G2 X148.76 Y2350. I40.55 J397.94
N13 G1 X850.
N14 Y850.
N15 X2150.
N16 Y2350.
N17 X2851.238
N18 G2 X2494.93 Y2002.06 I-396.86 J50.
N19 G1 X2450. Y1997.482
N20 Y550.
N21 X1650.
N22 Y-50.
N23 X3050.
N24 Y1008.093
N25 X3454.376 Y1675.313
N26 Y2155.923
N27 X3809.865 Y2584.88
```

Figura 5.22 Código G y M generado en Mastercam.

Una vez que se generó el código, se depura manualmente, eliminando los códigos que no son reconocidos por nuestro sistema.

5.5.3 Ejecución de un Programa CNC

Para la ejecución del programa CNC depurado, las opciones que se tienen con la interfaz desarrollada, son: copiar todas las líneas de código y pegarlas a el cuadro de texto denominado PROGRAMA CNC, o la otra opción es abrir el archivo (dando *click* al botón *Abrir*) guardado en alguna localidad de la PC.

Esta interfaz permite simular el programa y obtener gráficamente las trayectorias del resultado de maquinado detectando posibles situaciones erróneas. En la figura 5.23 se muestra la pantalla correspondiente a la simulación del proceso de maquinado del logo de los pumas de 4.73 x 4.83 mm.

Una vez realizada la simulación, y antes de ejecutar el programa es necesario definir el cero pieza, punto a partir del cual se harán referencia todas las coordenadas durante la ejecución del programa.



Figura 5.23 Simulación del programa CNC.

Finalmente obtenida la simulación correcta, se procede a ejecutar el programa de manera automática pulsando el botón *EJECUTAR PROGRAMA AUT.* En la figura 5.24 se ilustra el resultado del maquinado del escudo de los pumas.

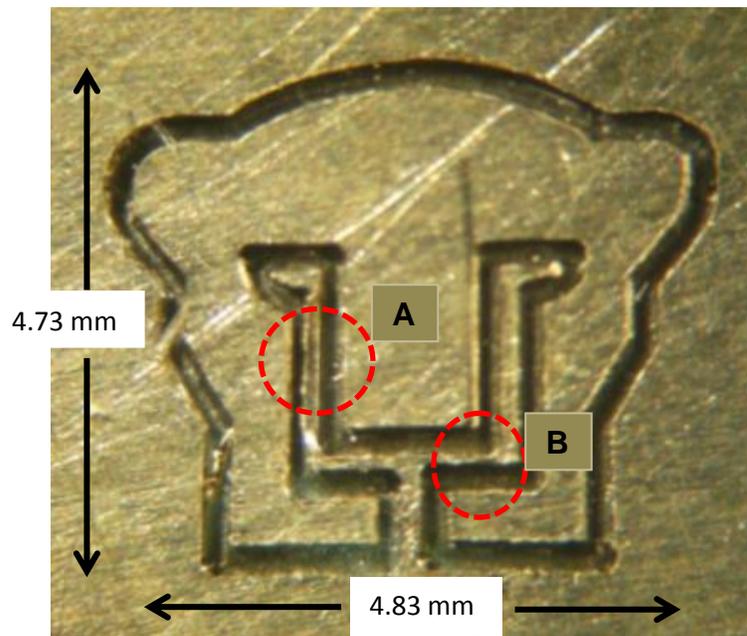


Figura 5.24 Maquinado del contorno del Puma.

En la figura 5.25 se muestra un acercamiento de las zona A y B correspondiente a la figura 5.24, en donde podemos apreciar que existen errores dimensionales de maquinado.

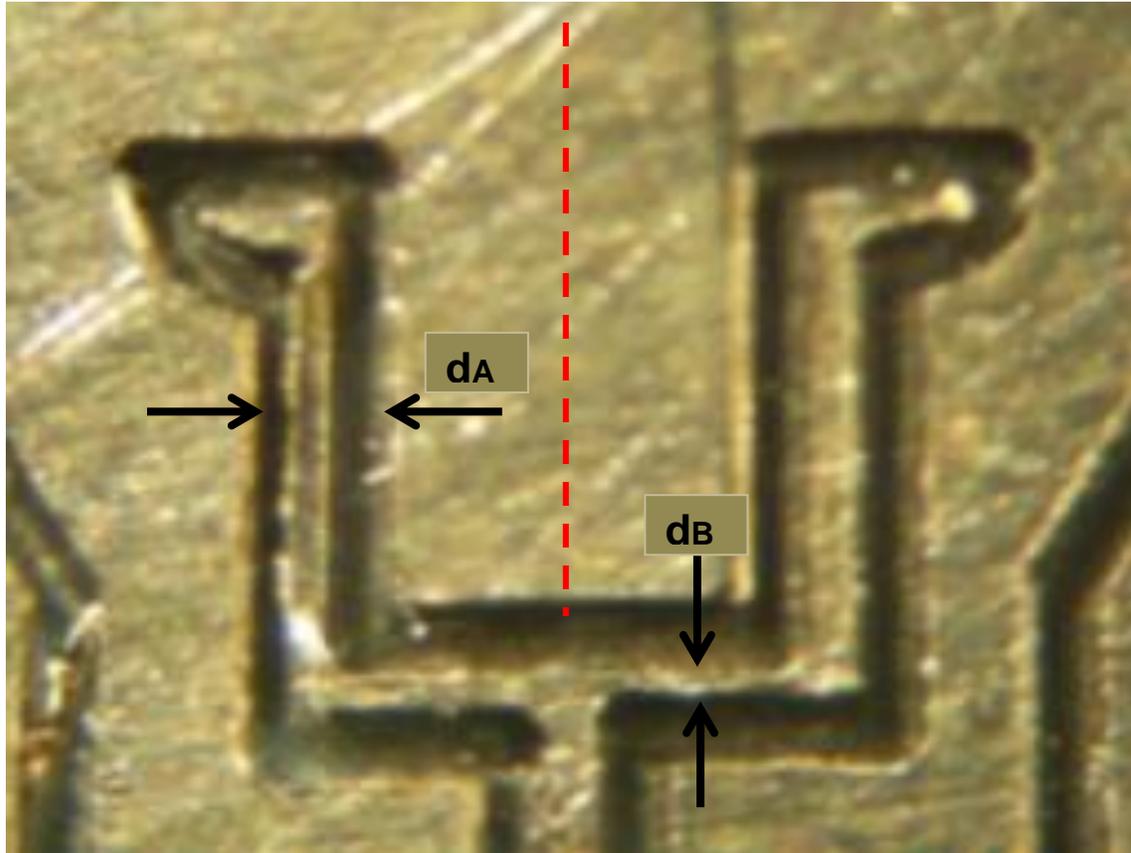


Figura 5.25 Errores presentes en maquinado de contorno del puma.

Medida	Dimensión teórica [μm]	Dimensión experimental
d_A	100	41
d_B	100	38

Tabla 5.17 Medición de los errores del puma.

Siguiendo el mismo procedimiento para el maquinado del contorno del puma se procedió a maquinar el logo del GMM con dimensiones en el eje X de 6 mm y en el eje Y de 5.26 mm. El resultado de dicho maquinado se muestra en la figura 5.26.

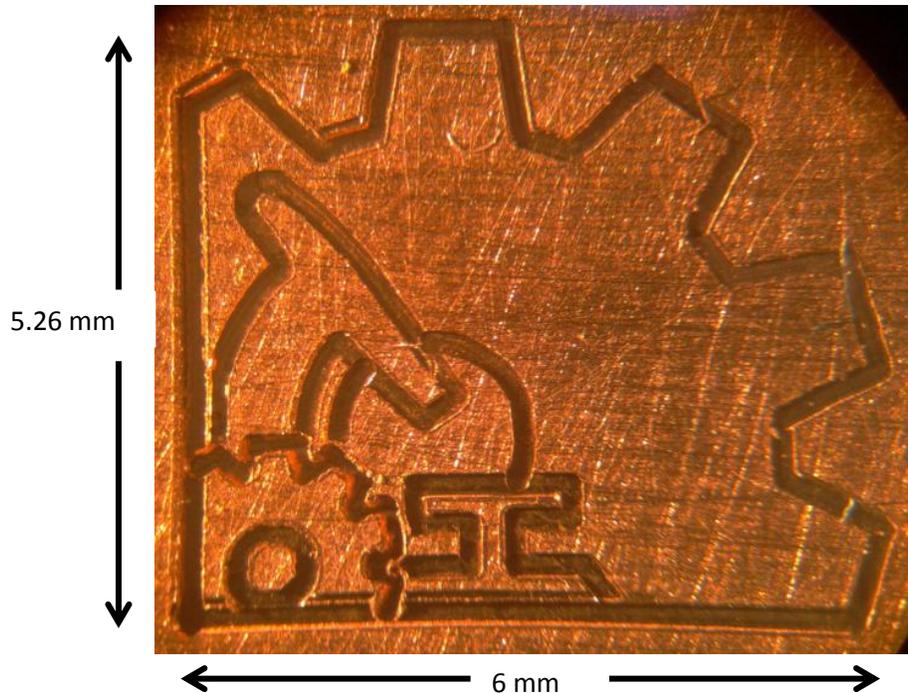


Figura 5.26 Maquinado del contorno del logo del GMM.

También se realizó el maquinado del contorno de un engrane con diámetro exterior de 1.5 mm. En la figura 5.27 podemos observar el resultado de este maquinado.

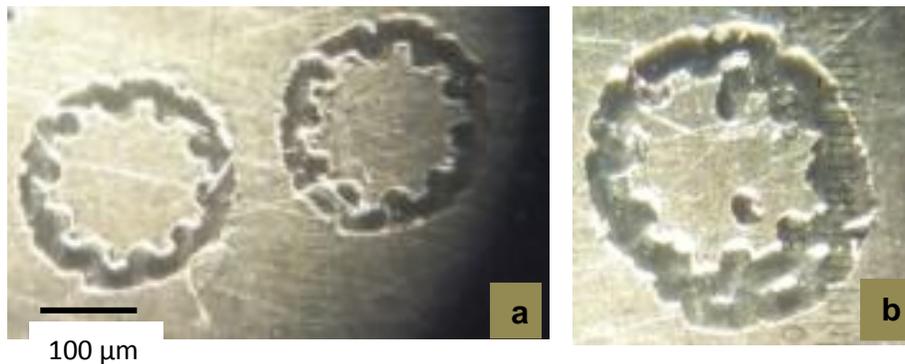


Figura 5.27 a) Maquinado de contorno de engrane en una pasada. b) Maquinado de contorno engrane en dos pasadas.

Con el resultado de los maquinados obtenidos hasta este momento, podemos deducir que aún existen errores debido a la compensación del juego mecánico presente en la micromáquina. Estos errores pueden reducirse cerrando el lazo al sistema de control.

5.5 Aplicaciones de las Micromáquinas Herramienta

Las micromáquinas herramienta tienen numerosas aplicaciones, pero es necesario conocer las limitaciones que trae consigo su operación en cuanto a precisión. Una de las restricciones presentes a la hora de manufacturar una pieza es la capacidad de la máquina de producir piezas con determinada exactitud.

De acuerdo con los resultados obtenidos en los maquinados realizados para el caso en el que se tienen movimientos en un solo eje a la vez, una de las aplicaciones potenciales, es en el área de microfluidos específicamente en maquinados de microcanales para el desarrollo de dispositivos microfluídicos. En la figura 5.28 se ilustra un esquema de flujo electro-osmótico, que consiste en el transporte de fluido a través de un microcanal, utilizando materiales que al entrar en contacto con una solución acuosa, se lleva a cabo una reacción química instantánea donde la superficie del material adquiere carga eléctrica.

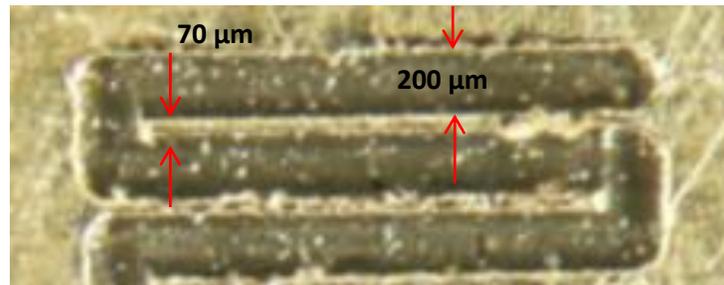


Figura 5.27 Microcanal de placas planas paralelas.

En máquinas de alta precisión, la manufactura de micropiezas, en diferentes materiales, como son: metales, semiconductores, plásticos y cerámicos. Los campos de aplicación principal son: la biomedicina, Automotriz, Electrónica, Aeroespacial, Micromecánica, algunas micropiezas fabricadas se ilustran en la figura 5.28.



Figura 5.28 (a) Micropinza de cerámica menor de 3 mm para aneurisma, con tolerancia de 20 μ m. (b) Micromolde para brackets dentales de 0.1 a 0.2 mm.

CONCLUSIONES

Al finalizar este trabajo, se concluye lo siguiente: de las diferentes técnicas de control empleadas en los sistemas CNC, la técnica de control empleada para el sistema de control fue la *Técnica por referencia de pulsos*. Se analizaron y evaluaron tres algoritmos de interpolación (DDA, Escalera y DSM), para el desarrollo de geometrías con trayectorias rectas y circulares, los métodos seleccionados fueron: el algoritmo por DDA para segmentos rectilíneos, y el algoritmo por búsqueda directa (DSM) para trayectorias circulares.

Para la integración de la tarjeta de control, se llevó a cabo la comparación y selección de componentes electrónicos, el driver de potencia seleccionado fue el R-325 y el microcontrolador PIC para realizar las tareas de interpolación y cálculos de parámetros fue el PIC 18F4550.

Para la programación de códigos G y M con el sistema de control implementado son 8 códigos G que se reconocen, los cuales consisten en movimientos de marcha rápida (G00 y G28), interpolación lineal y circular (G01, G02 y G03), retardos (G04) y sistema de referencia (G90 y G54), con estos códigos es suficiente para realizar procesos de microfresado.

Se desarrolló una interfaz gráfica utilizando el software Visual Studio C#. Con esta interfaz nos permite operar la MMH en cuatro modos: por *joystick*, por pasos, ejecutar línea por línea de código, y ejecutar un programa CNC de manera automática. La PC tiene la función de monitorear y enviar cadenas de control, a través de comunicación serial RS-232, cada vez que se finaliza de ejecutar una instrucción en la MMH. Otra de las funciones de la PC son: simular el proceso de manufactura, logrando una representación gráfica de las trayectorias de maquinado, mostrarlas coordenadas referentes al cero de máquina y al cero de pieza, así como también la velocidad de avance.

Finalmente se realizaron pruebas con dos micromáquinas, una con la MMH tipo puente y la otra con la segunda versión del primer prototipo: las primeras pruebas consistieron en determinar la frecuencia máxima a la que operaran los actuadores, la segunda prueba consistió en realizar maquinados de rectas paralelas para cuantificar el juego mecánico existente en cada uno de los ejes, los valores de compensación para cada eje fueron los siguientes: para el *Eje X positivo* fue de 314 μm , en el *eje X negativo* se compensó con 305.3 μm , para el *eje Y positivo* la compensación fue de 264.5 μm y para el *eje Y negativo* 273.8 μm . Posteriormente se llevaron a cabo maquinados realizando interpolaciones circulares, y por último la ejecución de programas de manera automática para el maquinado del contorno del logo de los pumas y del GMM.

Una limitante en la ejecución de programas CNC, es el tiempo requerido para realizar un maquinado, este factor lo determina la frecuencia máxima a la que operan los motores, la cual corresponde a 39 KHz. Relativamente esta frecuencia es alta, pero con el uso de micropasos está limitada para operar a una velocidad máxima de 14 mm/ min en la MMH tipo puente y 4 mm/min con la segunda versión del primer prototipo.

TRABAJO A FUTURO.

Dentro de la propuesta presentada en el presente trabajo se pueden mejorar algunas cuestiones:

En cuanto a la tarjeta controladora, la transmisión de los trenes de pulsos a los motores de pasos reduce velocidad en los ejes, por lo que puede mejorarse si se establece en la tarjeta de controlar la instalación de un microcontrolador (gobernados por un maestro) por cada eje de trabajo y así optimizar el tiempo de operación del proceso.

En cuanto a la comunicación entre la PC y tarjeta de control se recomienda cambiar por otro protocolo de envío y recepción de datos, por ejemplo, la comunicación por USB (*Universal Serial Bus*) con el fin de incrementar la velocidad en la transmisión de datos para el proceso de manufactura.

En lo que respecta a los errores por compensación del juego mecánico y otros errores atribuidos al sistema de transmisión de engranes y tornillos guía, se recomienda ampliamente cerrar el lazo al sistema de control con el uso de encoders lineales o rotacionales, con la finalidad de mejorar y asegurar la precisión de las trayectorias en los maquinados.

Es necesario también que el sistema desarrollado posea la versatilidad y configuración para realizar operaciones de un centro de maquinado y de torno, con el fin de ampliar las aplicaciones a proceso de taladrado, machueado, producción de piezas cilíndricas en esta MMH.

En cuanto al diseño y construcción de micromáquinas herramienta es necesario contar con una estructura rígida en la máquina para disminuir errores debidos a los efectos de vibración, se recomienda utilizar mecanismos antibacklash.

REFERENCIAS

- [1] Li. Jianjun, Li Xibing, Liu Jun-an, Wei Kxiang. *Geometric error analysis for millimeter-scale machine tool*.
- [2] Kussul, E., Baydik, T., Ruiz, L., Caballero, A., Velazco, G. *Development of Low-cost Microequipment*. 2002 IEEE International Symposium on Micromechatronics and Human Science. Nagoya Municipal Industrial Research Institute (Eds.), Nagoya, Japan.
- [3]. Yuichi Okasaki, NozumoMishima, and KiwamuAshida. *Microfactory-Concept, History and Development*. Journal of Manufacturing Science and Engineering. 2004.
- [4] Alberto Caballero-Ruiz, Leopoldo Ruiz-Huerta, Tatiana Baidik, Ernst Kussul. *Geometrical error anlysis of a CNC micro-machine tool*.
- [5] E. Kussul, T. Baidik, L. Ruiz-Huerta, A. Caballero, G. Velasco, and L. Kasatkina. *Development of Micromachine tool prototypes for microfactories*. Journal of Micromechanical and Microengineer. October 2002.
- [6] C.R. Friedrich and M.J. Vasile. *Development of the micromilling process for hight- aspect-ratio microstructures*. Journal of Microelectromechanical Systems. 5:33-38, 1996.
- [7] N. Ooyama, S. Kokaji, M. Tanaka, and et al. *Desktop machining microfactory*. In Proceedings of the 2nd International Workshop on Microfactories, pages 14-17. Switzerland, October 2000.
- [8] E. Kussul, D. Rachkovskij, T. Baydik and S. Talayev. *Micromechanical engineering: a basis for the low cost manufacturing of mechanical microdevices using microequipment*. 1996.
- [9] Leopoldo Ruiz Huerta. *Desarrollo de microequipo para células de manufactura*. División de estudios de posgrado, Facultad de Ingeniería. Abril 2005. Tesis de Doctorado, Ingeniería Mecánica.
- [10] 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.

[11] Weck, Manfred. *Handbook of Machine Tools, Vol. 1,2 y 3*. John Wiley and Sons. 1984.

[12] Leopoldo Ruiz Huerta, Alberto Caballero Ruiz. “*Presentación del GMM*”, 2006-2007, CCADET, UNAM.

[13] D. Dornfeld, Min,S., and Takeuchi Y. *Recent Advances in Mechanical Micromachining*. Manufacturing technology Annals of the CIRP Annals, 55,2.2006

[14]<http://www.faulhaber-group.com/n626141/n.html>

[15] Alberto Caballero Ruiz. *Caracterización de un microcentro de maquinado de bajo costo*. Tesis de Maestría. Ingeniería Mecánica. Septiembre 2000.

[16] E. Brinksmeier, W. Preu, “*Fabrication of precision molds*”, First International Conference of the European Society for Precision Engineering and Nanotechnology, pp. 187-190, 1999.

[17] Kornel F. Ehmann, Richard E. Devor and Shiv G. Kapoor. *Micro/Meso-scale Mechanical Manufacturing – Opportunities and Challenges-*.

[18] Okazaki, Kitahara. *Micro-machine Tool to machine micro-parts*. Proc. ASPE Annual meeting, 2000.

[19] E. M. Kussul, L. M. Kasatkina. *Los problemas de desarrollo de máquinas herramienta controladas con computadora para fábricas micro mecánicas sobremesa*. Sistemas de control y computadoras. 1998, N5, V.28, pp 32-39.

[20] 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.

[21] Alberto Caballero Ruiz. *Metodología de evaluación para microequipo*. División de estudios de posgrado, Facultad de Ingeniería, 2005. Tesis de doctorado. Área mecánica.

[22] 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.

- [23] Alberto Caballero R. Leopoldo Ruiz H. *Apuntes de diseño y construcción de prototipos*. Programa de estudios de maestría, periodo escolar 2005-2. Ciudad Universitaria.
- [24] Héctor Hugo Silva López. *Desarrollo de un Sistema de Control para una Micromáquina Herramienta Basado en Código Estándar para CNC*. División de Estudios de Posgrado. Facultad de Ingeniería 2008. Tesis de Maestría. Ingeniería Eléctrica.
- [25] 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.
- [26] Kalpakjian, Serop; Schmid, Steven R. *"Manufactura: Ingeniería y Tecnología"*. 4ª Edición Prentice Hall, México, 2002.
- [27] Ferré Masip R. *Como programar un control numérico*. Editorial Alfaomega. México, D.F. 1999. pags 155.
- [28] L. Huang, N. He and L. Li. *Study on an Open CNC System of 3-axis Micro Milling Machine Tool*. *Advanced Materials Research* Vols. 69-70 (2009) pp 639-373.
- [29] Gowri S., Ranjith P., Vijayaraj R., Balan A.S.S. *Micromachining: Technology of the future*. *Journal of Materials and Structural Integrity*, volume 1. 2007.
- [30] Industrial Centre, the Hong Kong Polytechnic University. *Computer Numerical Control (CNC)*. IC Professional Training Series.
- [31] Smid P. *A Comprehensive Guide to Practical CNC Programming*. *CNC PROGRAMMING HANDBOOK*. Editorial Industrial Press, Inc. Segunda edición. Nueva York, EUA, 2003.
- [32] Ricardo Jiménez. *Control Numérico por Computadora (CNC)*. Ingeniería de Manufactura, 2002.
- [33] 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.

[34] 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.

[35] Y, Koren. *Computer Control of Manufacturing Systems*. Technion Israel Institute of Technology. Editorial McGraw-Hill, Inc. México, 2003.

[36] 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.

[37]http://www.mech.mcmaster.ca/undergraduate/courses/me3m02/pdf_files/me302_expt_p3.pdf. Experiment 3 CNC Machining.

[38] Suk-Hwan, Seong-Kyoon Kang, Dae-Hyuk Chung, Ian Stroud. *Theory and Design of CNC System*.

[39] Atila Ertas, Jesse C. Jones. *The Engineering Design Process*. Department of Mechanical Engineering Texas Tech University. Editorial JhonWiley&Sons, Inc. 1993.

[40] 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 Ingeniería mecánica electricista.

[41] KhajaShoaibMoin. *Report on a CNC machining Centre and its Applications*. University of Huddersfiels. School of Computing and Engineering.

[42] German Herrera Granados. *Investigación y Evaluación de Mecanismos para el Desarrollo de Micromáquinas Herramienta*. Facultad de Ingeniería. Ciudad Universitaria. Febrero 2010. Tesis de Maestría en Ingeniería mecánica.

[43]http://www.deprag.com/fileadmin/bilder_content/emedi/broschueren_pics/emedi_druckluftmotoren/D6000/D6000en.pdf

[44] Q. Y. Wang, F. Z. Fang, and X. T. Hu. *Development of three-axis desktop milling machine*. 26 May 2009.