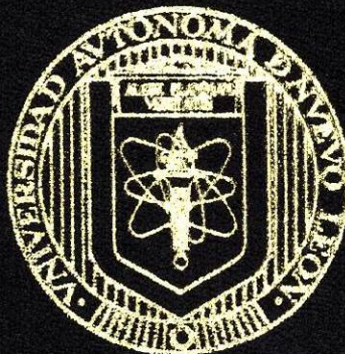


UNIVERSIDAD AUTONOMA DE NUEVO LEON  
FACULTAD DE INGENIERIA MECANICA Y ELECTRICA



DISEÑO DE UN CONTROL DE TRAYECTORIAS  
PARA UNA MESA DE CORTE CON LASER

TESIS

EN OPCION AL GRADO DE  
INGENIERO EN ELECTRONICA Y COMUNICACIONES

PRESENTA

JOSE SALVADOR ORTIZ CALDERON

ASESOR: M.C. JUAN ANGEL GARZA GARZA

DIRECTOR DE TESIS:

ASESOR: DR. VICTOR MANUEL COELLO CARDENAS (CICESE)

SAN NICOLAS DE LOS GARZA, N. L.

OCTUBRE DE 2003

DESIGNO DE UM CONTROL DE TRAJECTORIAS  
PARA UM A Mesa DE CORTE COM LASER

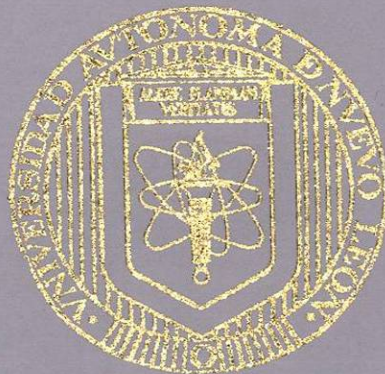
TL  
TA1675  
.078  
2003  
c.1



1080171466

Ingeniería y  
Tecnología

UNIVERSIDAD AUTONOMA DE NUEVO LEON  
FACULTAD DE INGENIERIA MECANICA Y ELECTRICA



DISEÑO DE UN CONTROL DE TRAYECTORIAS  
PARA UNA MESA DE CORTE CON LASER

TESIS

EN OPCION AL GRADO DE  
INGENIERO EN ELECTRONICA Y COMUNICACIONES

PRESENTA

JOSE SALVADOR ORTIZ CALDERON

ASESOR: M.C. JUAN ANGEL GARZA GARZA

DIRECTOR DE TESIS:

ASESOR: DR. VICTOR MANUEL COELLO CARDENAS (CICESE)

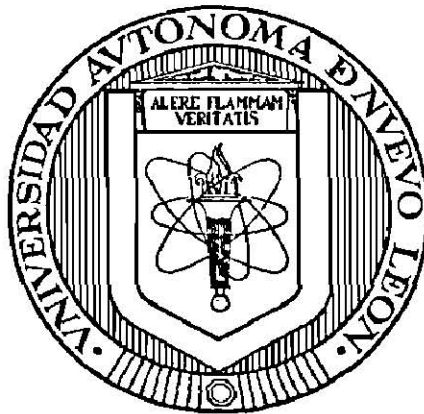
SAN NICOLAS DE LOS GARZA, N. L.

OCTUBRE DE 2003





**UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN**  
**FACULTAD DE INGENIERÍA MECÁNICA Y ELÉCTRICA**



**“DISEÑO DE UN CONTROL DE TRAYECTORIAS PARA UNA MESA  
DE CORTE CON LÁSER”**

**POR**

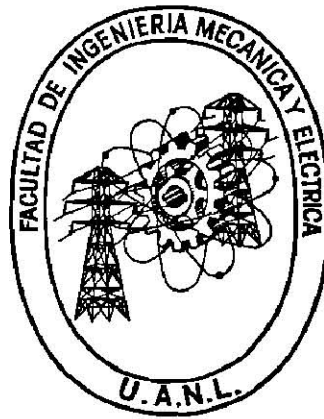
**JOSÉ SALVADOR ORTIZ CALDERÓN**

**TESIS**

**EN OPCIÓN AL GRADO DE INGENIERO EN ELECTRÓNICA Y  
COMUNICACIONES**

**SAN NICOLÁS DE LOS GARZA, N. L.  
OCTUBRE DEL 2003**

**UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN**  
**FACULTAD DE INGENIERÍA MECÁNICA Y ELÉCTRICA**



**“DISEÑO DE UN CONTROL DE TRAYECTORIAS PARA UNA MESA  
DE CORTE CON LÁSER”**

**POR**

**JOSÉ SALVADOR ORTIZ CALDERÓN**

**TESIS**

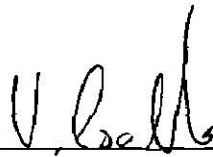
**EN OPCIÓN AL GRADO DE INGENIERO EN ELECTRÓNICA Y  
COMUNICACIONES**

**SAN NICOLÁS DE LOS GARZA, N. L.  
OCTUBRE DEL 2003**



# Resumen.

En esta tesis se hace un análisis para implementar un sistema de CNC en una mesa de trazos. Para esto se utilizan diversas herramientas matemáticas y diseño digital para el control de los motores de pasos y el desarrollo de trayectorias. Se hace uso de los PLDs para el diseño del controlador de los motores a pasos, asimismo se desarrolla el algoritmo para la generación de las trayectorias que seguirá la herramienta. Se utiliza como unidad de trayectoria la línea recta. El trabajo sirve como base para implementar un sistema de corte de materiales suaves usando un láser de CO<sub>2</sub>.



---

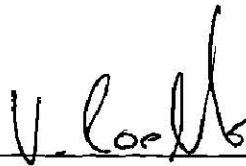
Dr. Víctor Coello

Director de tesis

**Palabras clave:** Láser, CNC, GALs, PLDs, HPGL, Motores a paso.

# Abstract.

This thesis reports on a study to implement a CNC control system in a plotter table. To deal with this, both, mathematical and digital designs tools are used for stepper motors control system as well as for the path designer. I made use of PLDs in order to design the control for stepper motors along with the algorithm for path generator. A straight line is used as a path unit. This work is the base for a cutting system of soft materials using a CO<sub>2</sub> laser.

A handwritten signature in black ink, appearing to read 'V. Coello', is positioned above a horizontal line.

Víctor Coello, Ph. D.

Supervisor

Keywords: Lasers, CNC, GALs, PLD's, HPGL, Steppers motors.

**Universidad Autónoma de Nuevo León.**  
**Facultad de Ingeniería Mecánica y Eléctrica.**

Recomiendo que la tesis **“DISEÑO DE UN CONTROL DE TRAYECTORIAS PARA UNA MESA DE CORTE CON LÁSER”**, realizada por el alumno José Salvador Ortiz Calderón, con matrícula 969521 sea aceptada para su defensa como opción al grado de Ingeniero en Electrónica y Comunicaciones.



---

Asesor

M. C. Juan Ángel Garza Garza

# DEDICATORIA.

A quien sino a mi madre, que me a guiado en todo mi camino y me ha apoyado en todas mis decisiones.  
Gracias Madre.

# AGRADECIMIENTOS.

Primeramente quiero darle las gracias a Dios por haberme permitido dar término satisfactorio a uno de mis muchos objetivos trazados en la vida.

*Este trabajo es dedicado a mis padres: Sonia Estefanía Calderón Vega y Salvador Ortiz Zúñiga, por el apoyo y las enseñanzas que me han brindado a lo largo de mi vida. Gracias por enseñarme que nunca hay que darse por vencido.*

Muy especialmente le dedico a mi hermana, Liliana Evangelina Ortiz Calderón, este trabajo y le agradezco el apoyo brindado durante esta etapa de mi vida.

Gracias a todos mis amigos que estuvieron alentándome para finalizar mi meta: Luis Alberto Zúñiga Segovia, Ricardo Alejandro Ochoa Ríos, Xavier Alejandro Briones Cortés, Jorge Iván Serna Hernández y en especial a Amanda Oropeza Verástegui por haber estado conmigo en todo la etapa de desarrollo de mi tesis. Gracias especialmente a Alejandro Gilberto García Mendoza por haberme asesorado y además facilitado material y equipo para los experimentos realizados.

Gracias a todas aquellas personas que con su ayuda y sus consejos también me han ayudado en especial a mis asesores: Dr. Víctor Coello y M. C. Juan Ángel Garza Garza por su tiempo, apoyo y confianza y especialmente al Dr. Ricardo Villagómez Tamez por su paciencia, su ayuda y sus consejos.

Gracias al CICESE y al CONACYT por el apoyo que me brindaron a través del proyecto “Optimización de los parámetros de diseño en láseres de CO<sub>2</sub> de guía de onda excitadas por RF en configuración de tableta slab laser”. Número del proyecto: 35176 A.

# PROLOGO.

La innovación siempre va aunada al desarrollo. Para innovar es necesario contar con el conocimiento para poder aterrizar las ideas. En el presente estamos rodeados de sueños del pasado, sueños que se ha realizado gracias al ingenio del hombre y gracias a la ciencia. Ejemplos de estos sueños son el automóvil, grandes rascacielos, los aviones entre otros, que en el pasado ni siquiera imaginaban.

La ingeniería ha contribuido enormemente en el desarrollo y en la innovación de muchos productos y procesos. Por esto, lograr unir tanto el conocimiento como la práctica de esta disciplina, es el mayor reto de todo ingeniero.

El presente trabajo se ha desarrollado como un caso práctico en el que se ha implementado un sistema CNC a una mesa de trazos. Este sistema de CNC es capaz de realizar trayectorias con la mesa de trazos. En un futuro se espera la implementación de un láser de CO<sub>2</sub> como herramienta para realizar cortes de materiales suaves. El equipo originalmente contaba con los motores de pasos y las transmisiones necesarias para el movimiento. A partir de esto se implementó los controladores de los motores y los algoritmos para generar trayectorias.

Este trabajo va dirigido a los estudiantes cuya licenciatura es cursada en el área de la ingeniería en control, electrónica y sistemas, ya que aborda conceptos y aplicaciones interesantes dentro de estas áreas. Por lo tanto se espera que el presente estudio y desarrollo físico del mismo contribuya hacia todos aquellos interesados en estos temas.

# INDICE.

SÍNTESIS.....	1
1. INTRODUCCIÓN.....	5
1.1 Descripción del problema.....	5
1.2 Objetivo de la tesis.....	5
1.3 Límite de estudio.....	5
1.4 Justificación del trabajo de tesis.....	5
1.5 Metodología.....	6
1.6 Revisión bibliográfica.....	6
2.1 FUNDAMENTOS Y TERMINOLOGÍA.....	8
2.1 Introducción.....	8
2.2 Motores a paso: descripción y sus principales características.....	8
2.2.1 Motores de imán permanentes.....	9
2.2.2 Torque de posesión y restauración.....	10
2.2.3 Características estáticas.....	11
2.2.4 Características dinámicas.....	12
2.2.5 Resolución.....	13
2.3 Láseres.....	14
2.3.1 El proceso de emisión láser.....	14
2.3.2 Requisitos fundamentales para la emisión láser.....	15
2.4 El proceso de corte por láser.....	17
2.5 El lenguaje HPGL.....	19
3. MODELADO DEL SISTEMA.....	24
3.1 Introducción.....	24
3.2 Datos de los componentes eléctricos y mecánicos.....	24
3.2.1 Especificaciones de los motores.....	25
3.2.2 Especificaciones del sistema reductor.....	25
3.3 Modelado del sistema.....	27

3.3.1 Constante del sistema.....	27
3.3.2 Velocidad constante.....	28
3.3.3 Aceleración constante.....	28
4. DESARROLLO E IMPLEMENTACIÓN DEL PROYECTO.....	30
4.1 Introducción.....	30
4.2 Diagrama general.....	30
4.3 Mesa de trazos.....	32
4.4 Controlador de los motores a pasos.....	34
4.5 Sistema generador de trayectorias (Computadora).....	38
5. DISEÑO DE LA INTERFASE.....	40
5.1 Introducción.....	40
5.2 Trazos de figuras.....	40
5.2.1 Punto.....	40
5.2.2 Rectas.....	43
5.2.3 Círculos.....	46
5.2.4 Archivos HPGL.....	48
6. CONCLUSIONES Y RECOMENDACIONES. ....	51
6.1 Conclusiones.....	51
6.2 Recomendaciones.....	52
Referencias.....	53
Bibliografía.....	54
Listado de figuras.....	55
Anexos.....	57
Autobiografía.....	98



# SÍNTESIS

En la industria es ventajoso el uso de tecnología de punta para realizar los procesos de transformación de materia prima en productos terminados. Es decir, se requiere la fabricación en serie de productos de alta calidad en tiempo corto. Un ejemplo es el corte y grabado de materiales con láser. Los cortes y grabados producidos por medio de este proceso son más limpios y más exactos debido a las propiedades del láser. Esto se complementa con las técnicas de CNC (Computer Numerical Control, por sus siglas en inglés), que facilitan la creación de las trayectorias seguidas por el haz (spot) del láser. Gracias a este tipo de equipos se pueden realizar con rapidez productos con bajo costo en mano de obra y con calidad excelente.

Desde que a principios del siglo XX se postuló que la transición entre estados de energía al excitar un átomo puede emitir un fotón, hasta nuestros días, ha variado en gran medida el uso que se ha dado a la energía láser. El haz de luz coherente obtenido a la salida de un resonador óptico por excitación del medio activo es un haz láser (acrónimo de "Light Amplification by the Stimulated Emission of Radiation"), que debidamente tratado se convertirá en una fuente de energía de alto aprovechamiento. Las características propias de la energía láser posibilitan su utilización de manera directa en aplicaciones industriales de procesamiento de materiales en general [1].

El maquinado por láser es el proceso de remover el material mediante la interacción del haz del láser y el material a procesar [2]. Esto se logra mediante la transformación de la energía de un fotón que incide en el material a maquinado a energía térmica o energía fotoquímica, y remover el material fundiéndolo, por evaporación o ablación. Por otra parte, los procesos de maquinado tradicionales dependen del debilitamiento mecánico producido por la herramienta de corte para romper con las propiedades del material [2]. Las ventajas del proceso de maquinado por láser sobre los procesos de maquinados tradicionales son muchas. Entre estas ventajas destacan las siguientes [2]:

- No existe el contacto y siempre está libre de las fuerzas de reacción mecánica, mientras que en el maquinado tradicional existe el contacto mecánico directo y necesita de un mecanismo para balancear las fuerzas mecánicas. Al haber contacto con la pieza de trabajo la herramienta sufre de desgaste. Esto hace que después de producir varias piezas se vaya perdiendo precisión.
- Puede remover material en cantidades muy pequeñas. Por esta razón la línea de corte con láser es muy estrecha. Esto sirve para la fabricación de piezas muy pequeñas.
- La zona afectada por el calor es relativamente pequeña y la capa de solidificación es de dimensiones de micras. Por esta razón, la distorsión con este proceso es insignificante. En el maquinado tradicional, la distorsión en el área de trabajo es inevitable.
- Puede ser aplicado para todo material que pueda absorber la radiación del láser, mientras que en los procesos de maquinados tradicionales tienen que escoger una herramienta para cada material con diferente dureza.
- Puede realizar un nivel de calidad final de maquinado muy bueno en un solo proceso, mientras en el maquinado tradicional es necesario realizar mas de un proceso para obtener este nivel de calidad.
- Pequeños orificios invisibles, ranuras, texturas de superficie y grabados pueden ser realizados con alta calidad usando el láser. Por esto es usualmente mucho más económico y eficiente para las aplicaciones en micro escalas.
- Tiene la propiedad de ser más flexible. La luz láser puede ser transmitida y reflejada a diferentes lugares a alta velocidad. El posicionamiento 3D puede ser realizado fácilmente. La combinación de fibras ópticas para transmitir la energía láser y los robots pueden crear sistemas con una gran libertad dimensional.
- El maquinado láser es sensible a la propiedad de enfocamiento del haz del láser. Si el haz del láser se enfoca la intensidad es muy alta. La profundidad de enfoque del láser es relativamente pequeña (alrededor de 100 milímetros). La profundidad de corte es de este modo limitado debido a este factor.

- El uso del láser para la manufactura de piezas o productos es una técnica con muchas aplicaciones y hoy en día se siguen encontrando más de estas aplicaciones.

Al unir las propiedades del corte con láser con la tecnología del control numérico por computadora (CNC), se logra incrementar las propiedades de éste proceso.

Las ventajas que se obtienen son:

- La reducción de la mano de obra.
- Mayor precisión en el proceso.
- Capacidad de producir una pieza en serie.
- Mayor rapidez en el proceso.
- Ahorro de materia prima.

El costo de las máquinas herramienta que cuentan con CNC es muy elevado en comparación con los equipos que no lo tienen. Sin embargo, las ventajas antes mencionadas compensan la utilización de estos equipos.

El maquinado de curvas implícitas planas es muy común en el control numérico por computadora (CNC). Esto usualmente se lleva a cabo por aproximaciones de estas curvas, como segmentos de líneas rectas o arcos circulares, a través de las interpolaciones realizadas por el CNC. Las entidades geométricas de las curvas implícitas planas están relacionadas con las entidades del movimiento de la herramienta a lo largo de la curva [3].

El maquinado por CNC es un importante método para el maquinado de superficies. Este método envuelve 4 fases principales: optimización de los parámetros del proceso del maquinado, planeación de la trayectoria, interpolación de la trayectoria y el servo control [3]. Para la determinación de la velocidad el algoritmo de interpolación debe de tener en cuenta la optimización del proceso [3].

Debido a sus características para el control de la posición en el CNC es necesario de motores especiales llamados motores a paso. La característica principal de los motores a pasos es el hecho de poder moverlos un paso a la vez por cada pulso que se le aplique.

Además de controlar la cantidad de pasos también se puede controlar la velocidad variando la frecuencia de los pulsos que alimentan al motor.

En el CNC existen varios lenguajes para la generación de trayectorias. Uno de los lenguajes es el HPGL (lenguaje gráfico de Hewlett Packard). Este fue desarrollado por HP (Hewlett Packard) principalmente para el control de trazadores.

Los archivos HPGL, son archivos vectoriales, los cuales constan básicamente de:

- Coordenadas de inicio de línea.
- Coordenadas de final de línea.
- El atributo de color de la plumilla.
- El estado lógico de la plumilla.

El HPGL usa mnemónicos de dos letras como instrucciones para dibujar líneas, círculos, texto, y simples símbolos. Con este lenguaje se puede, en principio, crear cualquier tipo de trayectoria.

# CAPÍTULO 1

## INTRODUCCIÓN.

### 1.1 Descripción del problema.

Automatizar una mesa de trazos para el proceso de corte y grabado por láser mediante el control numérico por computadora (CNC). El control debe de realizar trayectorias geométricas básicas, además de poder realizar trayectorias escritas en lenguaje HPGL.

### 1.2 Objetivos de la tesis.

- Diseñar y construir un controlador electrónico para controlar motores de pasos en forma simultánea.
- Adaptar las funciones del sistema diseñado al control de movimientos de una mesa para trazos que cuenta con un láser para cortes de materiales suaves.
- Implementar las primitivas geométricas: Punto, recta y círculo.
- Implementar una interfase estandarizada de comandos (HPGL).

### 1.3 Límites del estudio.

Esta tesis se enfoca a diseñar un controlador adecuado para controlar e implementar las funciones básicas mencionadas en el punto anterior.

### 1.4 Justificación del trabajo de tesis.

La justificación de esta tesis se basa en los siguientes puntos:

- La necesidad de sistemas de CNC en la industria que optimicen las operaciones de corte y grabado con láser.
- La difícil adaptación de los sistemas de software ya existentes para el control de la mesa trazadora.
- El alto costo de los sistemas ya existentes.
- La necesidad de un diseño propio para su implementación y desarrollo.

## 1.5 Metodología.

En este trabajo se utilizan técnicas de diseño secuencial de electrónica digital para el desarrollo de los controladores de los motores a pasos. Para el desarrollo de los controladores se utiliza el GAL (por sus siglas en inglés, Generic Arrive Logic), que significa Arreglo Lógico Genérico, para la implementación del diseño. Todo esto con el fin de minimizar el costo y el tamaño del controlador.

En el desarrollo de la interfase de control, se utilizan métodos numéricos y matemáticos para la generación de trayectorias, dando inicio con la definición formal de punto, continuando con la definición e implementación de trayectorias sobre líneas rectas y círculos. Por último se implementa un lenguaje estándar, <sup>este</sup> lenguaje es el HPGL, desarrollado por HP (Hewlett Packard), para el manejo de trazadores.

## 1.6 Revisión bibliográfica.

### **Electrónica Industrial Moderna, 3ª. Edición.**

De aquí se seleccionaron algunos de los conceptos básicos descritos en el capítulo 2, particularmente el funcionamiento de un motor a pasos y el diseño a gran escala del controlador.

En este libro puede encontrarse gran información sobre electrónica industrial y diseños de dispositivos de potencia.

### **LASERS. Operación, equipo, uso y diseño, 1ª. Edición.**

De esta obra se seleccionan los conceptos básicos sobre el funcionamiento del láser y en especial del láser de CO<sub>2</sub>. Todo esto mencionado en la sección del láser del capítulo 2 de la presente tesis. En esta obra se encuentra además información técnica sobre el tratamiento y el procesado de varios tipos de materiales y se muestran algunos sistemas láser.

### **Manual de prácticas de laboratorio de Electrónica Lógica I, 1ª. Edición.**

De este manual de prácticas se obtuvo la información con respecto al GAL (Generic Array Logic), el lenguaje ABEL y además el diseño del controlador de los motores a paso.

**Física (Incluye física moderna), 3ª. Edición**

El libro abarca varios temas de la física. Este libro ~~me~~ sirvió para obtener los análisis de velocidades y aceleraciones del sistema en el capítulo 3

**Como programar en C/C++, 2ª Edición.**

El libro muestra de manera sencilla y completa el lenguaje C/C++. Abarca desde los conceptos básicos del lenguaje hasta estructuras de datos más complejas y manejo de archivos. Este libro sirvió como apoyo en el desarrollo del programa que controla las trayectorias de la mesa de trazos.

**Diseño Digital. Principios y Prácticas, 1ª Edición.**

Este es un libro de circuitos digitales muy completo y muy sencillo de comprender. Cuenta con una sección de dispositivos de lógica de arreglo programable (PAL) en la cual se habla del lenguaje ABEL del diseño de circuitos secuenciales por medio de PALs.

# CAPÍTULO 2

## FUNDAMENTOS Y TERMINOLOGÍA.

### 2.1 Introducción.

En una mesa para corte con láser los componentes principales para el movimiento y la generación de la trayectoria son los motores a paso. Estos motores tienen propiedades y características necesarias para cumplir con los requerimientos de posicionamiento y de velocidad requeridos en este contexto. El corte es realizado por medio de luz láser aprovechando las propiedades que ésta tiene para obtener cortes muy finos y precisos [2]. Para tener una idea más clara de las ventajas del uso del láser para el corte de materiales y los motores a paso para el posicionamiento y la generación de trayectoria es necesario conocer sus principios de funcionamiento y sus características.

### 2.2 Motores a paso. Descripción y sus principales características.

De manera general, se puede decir que los motores a pasos son mecanismos que convierten señales de pulsos eléctricos en un movimiento continuo del rotor hasta fijarlo en un determinado paso o posición. Estos motores se diseñan para tener un control de posición preciso y son estables en lazo abierto para cualquier posición del paso, sin embargo, tiene una respuesta al escalón con un sobreimpulso considerable y un tiempo de establecimiento relativamente grande. Aunque el control en lazo abierto es ventajoso económicamente, no está exento de limitaciones. Por ejemplo, las revoluciones del rotor se toman inestables en ciertos rangos de velocidad, y en consecuencia, la velocidad y la aceleración de un motor a pasos controlado bajo un esquema de lazo abierto no puede ser tan rápida como la de un motor de CD operado bajo un esquema de control en lazo cerrado.

Existen varios tipos de motores a pasos, entre los que podemos enumerar son los motores de reluctancia variable, los motores de magneto permanente y los denominados híbridos, siendo los dos primeros los más comunes.



Debido a que estos motores tienen posicionamiento preciso y confiable se emplean comúnmente en aplicaciones donde el posicionamiento mecánico resulta ser muy importante. Son ideales cuando se tiene que girar un eje y detenerlo en ciertas posiciones con una precisión de hasta centésimas de milímetros. Dicha posición permanece bloqueada hasta que no se dé una nueva orden para hacerlo girar en el sentido que uno quiera.

Es deseable que un motor a pasos se mueva tan rápido como sea posible en respuesta a un pulso de entrada o tren de pulsos. No solo se requiere de un rápido arranque, sino también de un rápido paro. Si el tren de pulsos es interrumpido mientras el motor está corriendo, el motor debe ser capaz de parar en la posición especificada por el último pulso. Estas excelentes características dinámicas se deben al alto índice de torque de inercia del rotor de un motor a pasos (comparado con un motor de CD).

### 2.2.1 Motores a paso de imán permanente.

Los motores a paso pertenecen a la clase de motores llamada motores de conmutación electrónica. Este tipo de motor es muy apropiado para mover el eje una cantidad de giro exacta.

Los motores a paso son fundamentalmente diferentes de los demás motores de CD ya que no tienen escobillas ni conmutador mecánico. En su lugar, la acción de conmutación necesaria para la función de motor de CD es lograda por transistores externos. Así el rotor es una colección de imanes permanentes salientes, como se muestra en la Figura 2.1.

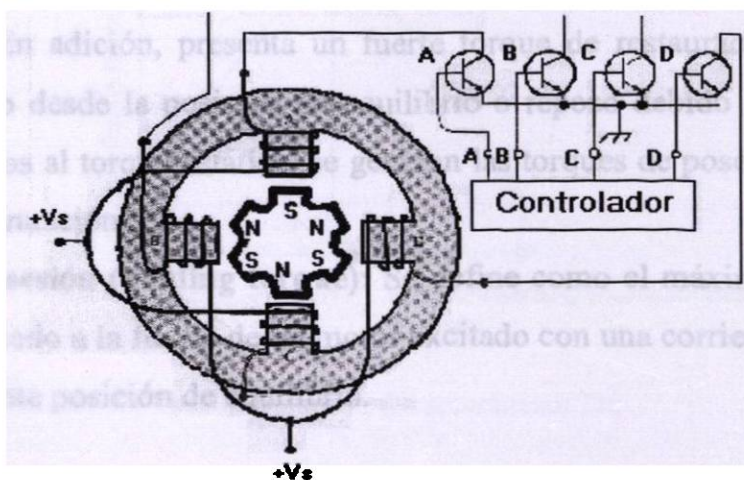


Figura 2.1 Motor a pasos.

Los cuatro devanados de los polos del estator y sus transistores controladores se etiquetan A, B, C y D. Cuando el circuito de control enciende un transistor en particular hay un flujo de corriente de alimentación de C. D.  $+V_s$ , a través de ese devanado en particular y a través del transistor a tierra. Cuando un solo devanado es energizado está enrollado de tal manera que su polo se vuelve norte magnético. Su flujo emerge de la cara del polo, pasa a través del rotor y completa su trayectoria entrando en la cara del directamente opuesto a él. Por ejemplo, si el transistor A energiza el polo A de la Figura 2.1, el flujo creado por ese polo completa su trayectoria a través del polo C y a través del armazón del motor. Por lo tanto, C automáticamente se vuelve un polo sur, aun cuando su devanado no lleva corriente.

En la Figura 2.1 los polos permanentes del rotor están etiquetados del 1 al 6, siendo sur los polos 1, 3 y 5. Los polos alternos 2, 4 y 6, son norte.

El principio de operación de un motor a paso de imán permanente es el siguiente: el polo del estator energizado que se vuelve norte magnético activo atrae el polo sur más cercano del rotor para alinearlo con él. Esta acción de producción de par es ayudada por el polo sur pasivo del estator (del lado opuesto del estator) atrayendo el rotor norte opuesto para alinearlo con él.

### 2.2.2 Torque de posesión y restauración.

Debido al alto torque estático que es generado, las propiedades de torque en un motor a pasos son muy interesantes. El torque estático se refiere al torque que se genera estando el motor en reposo. Este torque habilita al motor para arrancar y parar rápidamente. En adición, presenta un fuerte torque de restauración cuando ocurre un desplazamiento desde la posición de equilibrio o reposo debido a una carga de torque externa. Ligados al torque estático, se generan las torques de posesión y frenado, que se definen a continuación.

**Torque de posesión (holding torque):** Se define como el máximo torque estático que puede ser aplicado a la flecha de un motor excitado con una corriente sin causar rotación hacia la siguiente posición de equilibrio.

**Torque de frenado (detent torque):** Es el máximo torque estático que puede ser aplicado a la flecha de un motor no excitado sin causar rotación a la siguiente posición de equilibrio. Este torque solo aparece en motores que tienen un imán permanente.

En general, entre más grande es el torque de posesión, es más pequeño el error de posición debido a la carga o variaciones de la misma.

### 2.2.3 Características estáticas.

Cuando el motor se encuentra en estado estacionario o en una posición de equilibrio, se observan algunas características relacionadas con el torque, desplazamiento angular y corriente, llamadas características estáticas.

**Características  $T/\theta$ :** Si tenemos al motor a pasos mantenido en posición de equilibrio o estacionaria suministrando una corriente en un modo especificado de excitación, al aplicar un torque externo a la flecha un desplazamiento angular ocurrirá. La relación entre torque externo y el desplazamiento puede ser graficada como en la Figura 2.2. Esta curva es llamada curva característica  $T/\theta$ . El valor máximo de torque estático es llamado torque de posesión, el cual ocurre cuando  $\theta = \theta_M$  (Figura 2.2). Para los desplazamientos más grandes a  $\theta_M$ , el torque estático no actúa en una dirección hacia la posición de equilibrio original, pero lo hace en la dirección opuesta hacia la siguiente posición de equilibrio.

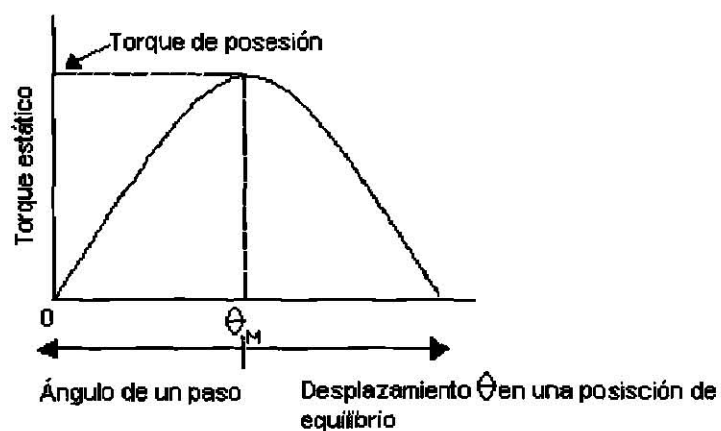


Figura 2.2. Características  $T/\theta$ .

**Características T/I:** El torque de posesión se incrementa con la corriente. Esta relación es convencionalmente referida como características T/I. Como un ejemplo, en la Figura 2.3 se presenta una comparación de esta característica con un motor a pasos híbrido y para uno de reluctancia variable (VR). El ángulo de paso de ambos es de  $1.8^\circ$  (resolución mínima de desplazamiento).

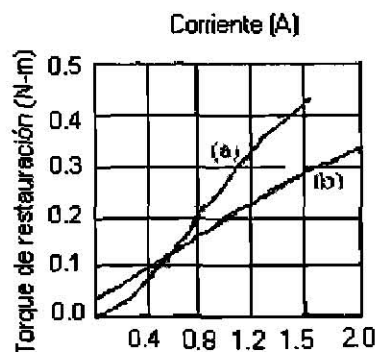


Figura 2.3 Características T/I.

## 2.2.4 Características dinámicas.

Las características relacionadas a los motores cuando están en movimiento o inmediatamente después de que arrancan, son llamadas características dinámicas (Figura 2.4).

**Características del torque de atracción (Pull in):** Son llamadas también características del arranque y se refieren al rango de torque de carga friccional al cual el motor puede arrancar o parar sin perder pasos o varias frecuencias en un tren de pulsos.

**Características del torque de arranque (Pull out):** También llamada, característica de ejecución, es la relación entre carga friccional de torque y máxima frecuencia de pulso con la cual el motor se puede sincronizar.

**Frecuencia máxima de arranque:** Esta definida como la máxima frecuencia de control a la cual un motor sin carga puede arrancar y parar sin perder pasos.

**Torque máximo de arranque:** También llamado torque máximo pull-out y es definido como la máxima carga de torque friccional con la cual el motor puede arrancar y sincronizar con un tren de pulsos de una frecuencia baja como 10 Hz.

**Máximo índice de arranque (Pull out):** Es la máxima velocidad a la cual un motor a pasos puede operar sin perder pasos.

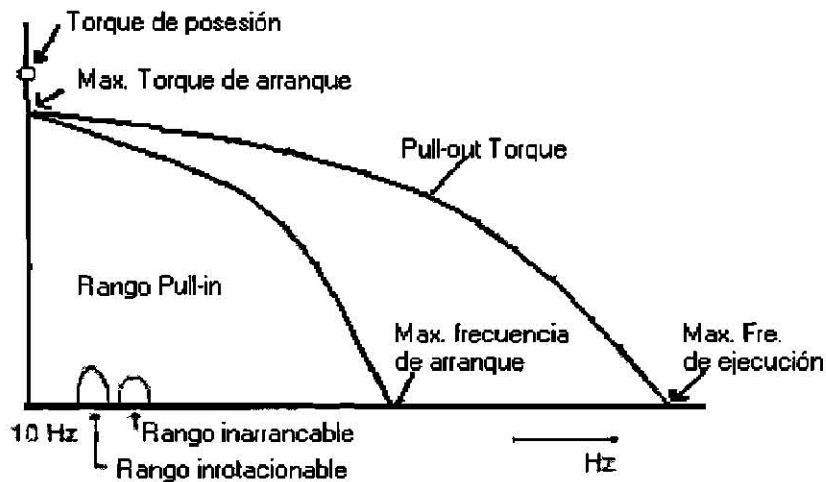


Figura 2.4. Características dinámicas.

### 2.2.5 Resolución.

Un motor a pasos rota a un ángulo fijo en cada pulso. Este desplazamiento es llamado paso y disminuyendo el ángulo de paso se incrementa la resolución de posicionamiento. Una característica de los motores a pasos es que pueden ser hechos para un ángulo de paso pequeño. Para aplicaciones en el área de ingeniería lo que nos interesa de estos motores es el número de pasos por revolución, los cuales están denotados por  $S$  y el ángulo de paso por  $\theta_s$ , cuya relación con  $S$  es:

$$\theta_s = \frac{360^\circ}{S} \quad \text{ec. (2.1)}$$

Donde  $S$  está relacionado al número de dientes en el rotor  $N_r$  y al número de fases  $m$ . La resolución se calcula:

Para el caso de motores VR:

$$S = mN_r, \quad \text{ec. (2.2)}$$

para el caso de motores de magneto permanente o híbridos:

$$S = 2mN_r. \quad \text{ec. (2.3)}$$

Así podemos calcular cual es la resolución límite, es decir, el mínimo ángulo de paso del motor que se este utilizando.

## 2.3 Láseres.

La proposición de que las partículas de luz con energía de una frecuencia específica podrían estimular a los electrones del átomo para emitir energía radiante, como luz de la misma frecuencia, se debe a Albert Einstein en 1917. En este fenómeno se basa el funcionamiento del láser. El nombre "láser" es un acrónimo de light amplification by stimulated emission of radiation (amplificación de la luz por la emisión estimulada de radiación).

Fue en 1958 cuando C. H. Townes y A. L. Shawlow propusieron que los principios en que se basa la amplificación de microondas por emisión estimulada, para producir el maser, podrían aplicarse a la amplificación de la luz. Pocos años mas tarde, T. H. Maiman desarrolló el primer láser operante.

El desarrollo del primer láser de bióxido de carbono ( $\text{CO}_2$ ), se le acredita a C. K. N. Patel, quien realizó su trabajo en los laboratorios Bell. Apenas tres años después, en 1966, el primer láser industrial de  $\text{CO}_2$  fue construido por los ingenieros de Coherent Radiation Laboratories.

### 2.3.1 El proceso de emisión láser.

El proceso de la emisión láser podría expresarse en una forma simplificada como "lo opuesto a la absorción". Conforme a la mecánica cuántica, los átomos y las moléculas tienen niveles definidos de energía y pueden pasar de un nivel a otro en saltos discontinuos. El cambio de energía necesario para un salto, es dado por la absorción o la emisión de una cantidad discreta de radiación electromagnética. Las frecuencias de la radiación y los intervalos entre los niveles de energía son característicos del átomo y por tanto difieren de elemento a elemento.

En condiciones normales, la mayoría de los átomos o moléculas permanecen en su nivel más bajo de energía o estado fundamental. Cuando estas partículas son excitadas por un destello de luz, por una descarga eléctrica o por otros medios y pasan a un nivel de energía superior, al regresar al estado fundamental emiten luz incoherente. En una cavidad láser, tales fotones emitidos quedan atrapados entre espejos paralelos y altamente pulidos, forzándolos a reflejarse hacia delante y atrás dentro de la cavidad. Siempre que un fotón pase cerca de otra partícula excitada con la misma energía, esta segunda partícula será estimulada para que emita un fotón, que será idéntico en longitud de onda, fase y coherencia espacial al primero. Ambos fotones son ahora capaces de estimular la emisión de más fotones semejantes a ellos mismos y éstos también formarán parte de la cantidad creciente de fotones entre los espejos (Figura 2.5).

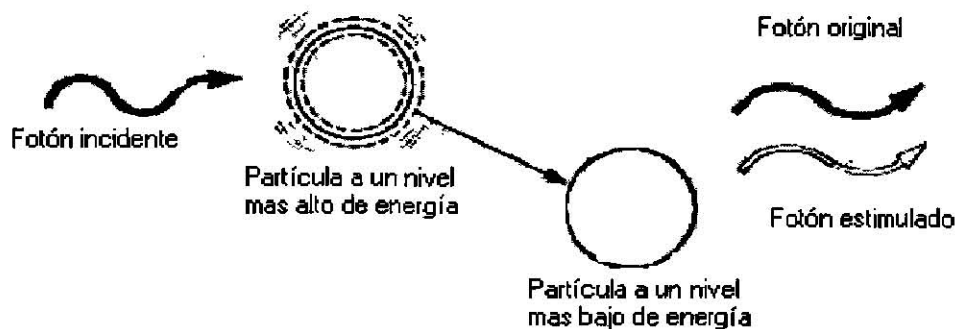


Figura 2.5. Emisión estimulada

La emisión láser empieza cuando hay suficientes fotones dentro de la barra. Si uno de los espejos es parcialmente transparente se emite un haz de luz coherente intenso y altamente ordenado.

### 2.3.2 Requisitos fundamentales para la emisión láser.

Todos los equipos láser tienen tres elementos fundamentales: la sustancia emisora, que proporciona átomos, iones o moléculas que producen la amplificación de la luz; una fuente de energía para excitar el medio y un resonador óptico para facilitar la retroalimentación de la luz que se amplifica. La Figura 2.6 ilustra en forma esquemática los elementos básicos empleados en el láser de bióxido de carbono (CO<sub>2</sub>).

No todas las sustancias tienen las propiedades para ser medios emisores láser, puesto que un medio eficaz de emisión láser debe ser suficientemente excitable para alcanzar el estado que se conoce como “inversión de población”. En esta condición hay una ganancia neta en la luz que se genera. Es decir, el medio excitado produce más fotones de los que absorbe.

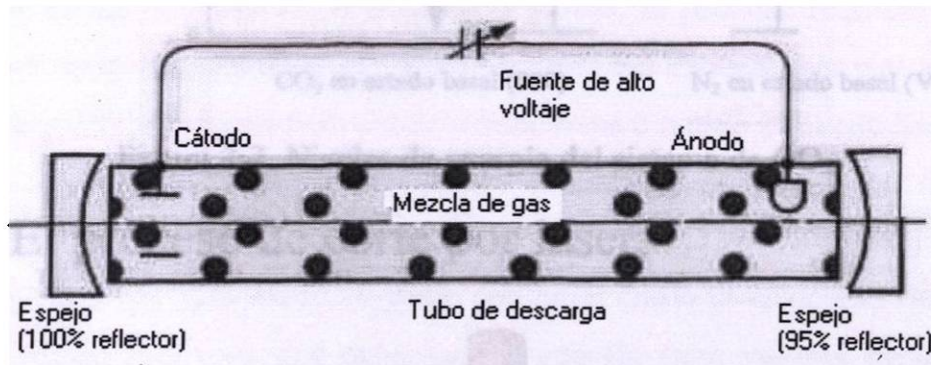


Figura 2.6 Elementos básicos del láser de CO<sub>2</sub>

En la Figura 2.7, la excitación comienza a la derecha con las moléculas de nitrógeno (N<sub>2</sub>) y se transfieren a las moléculas de CO<sub>2</sub> de la parte superior izquierda. De los niveles elevados de energía, las moléculas de CO<sub>2</sub> descienden emitiendo energía de 10.6 y 9.6 micras de longitud de onda. Existe una inversión de población, puesto que ahora hay más partículas en niveles de energía elevados (marcados con 100 y 020) que las que hay en los niveles inferiores. Esta proporción es importante porque únicamente las partículas que se encuentran en los niveles inferiores son capaces de absorber energía luminosa. Sólo las partículas que están en los niveles superiores contribuyen con fotones para el sistema.



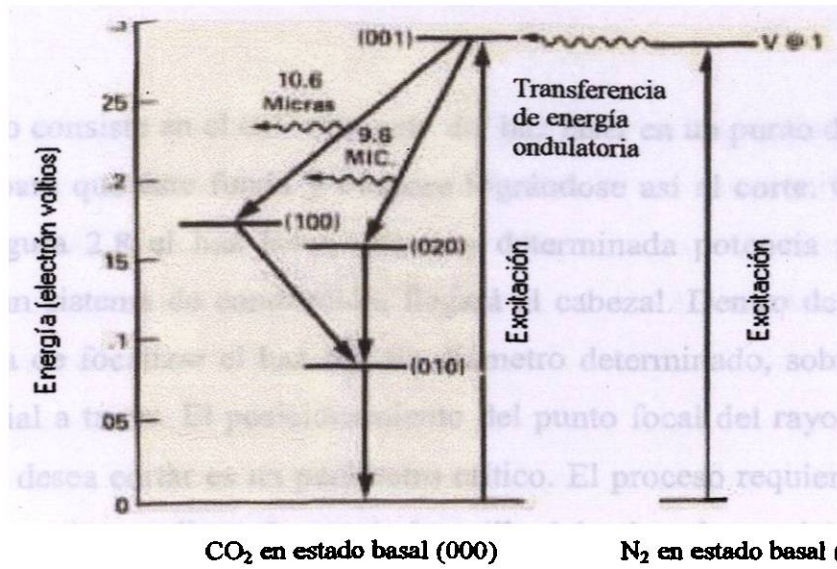


Figura 2.7. Niveles de energía del sistema de CO<sub>2</sub>

## 2.4 El proceso de corte por láser.

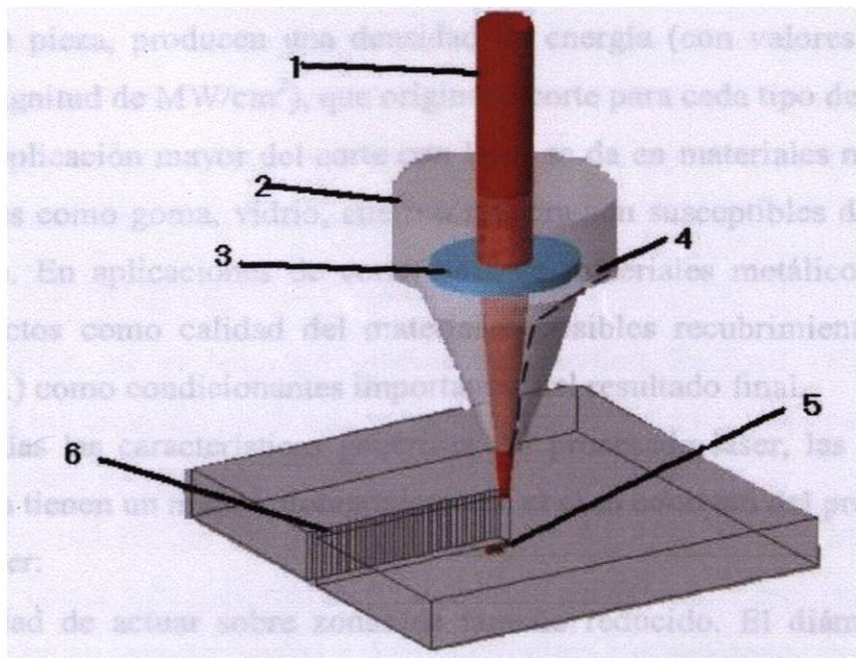


Figura 2.8. Modelado representativo del *proceso de corte*.

1. Luz láser procedente del generador y de un sistema de conducción.
2. Cabezal. 3. Sistema óptico. 4. Boquilla del cabezal 5. Zona de corte
6. Estrias o rugosidades en las superficies cortadas.

El proceso consiste en el enfocamiento del haz láser en un punto del material que se desea tratar, para que éste funda y evapore lográndose así el corte. Como **pretende** simbolizar la Figura 2.8 el haz láser, con una determinada potencia procedente del generador y de un sistema de conducción, llegará al cabezal. Dentro de él, un sistema óptico se encarga de focalizar el haz con un diámetro determinado, sobre un punto de interés del material a tratar. El posicionamiento del punto focal del rayo respecto de la superficie que se desea cortar es un parámetro crítico. El proceso requiere de un gas de asistencia, que se aplica mediante la propia boquilla del cabezal, coaxial al propio rayo láser. Este gas puede ser inerte para evitar oxidaciones o activo para catalizar el proceso. A su vez favorece la eliminación de material fundido, vapor y plasma de la zona de corte. Es típica la aparición de ciertas estrías o rugosidades en las superficies cortadas. La conjunción de todos estos factores, junto con otros como la velocidad relativa entre el cabezal y la pieza, producen una densidad de energía (con valores característicos en orden de magnitud de  $MW/cm^2$ ), que origina el corte para cada tipo de material.

Aunque la aplicación mayor del corte con láser se da en materiales metálicos, otro tipo de materiales como goma, vidrio, cuero o madera son susceptibles de ser cortados con este método. En aplicaciones de corte láser de materiales metálicos debe tenerse en cuenta aspectos como calidad del material o posibles recubrimientos (aceite, óxido, pinturas, etc.) como condicionantes importantes del resultado final.

De entre todas las características genéricas del procesado láser, las que se exponen a continuación tienen un mayor protagonismo en el caso concreto del proceso industrial de corte por láser:

- Posibilidad de actuar sobre zonas de tamaño reducido. El diámetro del spot que incide sobre la superficie a cortar tiene un valor medio en torno a las tres décimas de milímetro. Esto acarrea la consecución de surcos de corte muy estrechos de dimensiones muy parecidas a las del propio spot o ligeramente superiores. Asimismo, las distorsiones que origina en el material son mínimas.

- **Accesibilidad.** La posibilidad de transmitir el haz láser mediante espejos hace que se pueda alcanzar cualquier orientación de corte dentro del campo de trabajo.
- **No contacto mecánico con la pieza.** No se produce desgaste de la herramienta por contacto ya que el grupo óptico que enfoca el haz origina que en posición de trabajo exista una separación entre la boquilla de la que sale el rayo y la pieza.
- **Sistemas sofisticados.** La programación se hace de una forma cómoda y precisa. Los dispositivos pueden incluir tablas de parámetros para cortar diferentes materiales. Es posible la automatización del proceso así como la comunicación de la máquina láser con otro tipo de dispositivos como CNC, centros de procesado, elementos de control de calidad, sistemas de gestión de errores y alarmas así como dispositivos de monitorización on-line de la máquina y del proceso láser.

La gran desventaja que presenta el corte por láser frente a otros procedimientos reside principalmente en el espesor máximo que se puede cortar. Otros procedimientos como el oxicorte, corte por plasma, electroerosión o corte por chorro de agua permiten cortar espesores mayores que el láser. Solamente el corte por punzonado tiene la limitación de cortar espesores menores que los que corta el láser.

## 2.5 El lenguaje HPGL.

El HPGL es el lenguaje gráfico de Hewlett Packard (HP), diseñado para especificar la información gráfica para los trazadores. Este lenguaje es capaz de describir gráficos sencillos, es eficiente, fácil de escribir y leer.

Los comandos del HPGL son mnemónicos que consisten de dos letras en mayúscula seguidos por uno o más parámetros y finalmente terminado con el carácter separador el cual consiste en un punto y coma (;). Los parámetros son separados por comas (,).

El HPGL usa mnemónicos de dos letras como instrucciones para dibujar líneas, círculos, texto, y simples símbolos, la siguiente lista muestra algunos ejemplos:

¿Tabla?

COMANDOS DE LA INICIALIZACIÓN	
IN;	Indica el inicio del archivo. Prepara al controlador para la ejecución de los comandos HPGL con sus respectivas
COMANDOS DEL MOVIMIENTO	
PA[X, y][, z][[ x, y][, z ]... ]	Indica al controlador que trabaje con coordenadas absolutas.
PR[X, y][, z][[ x, y][, z ]... ]	Indica al controlador que trabaje con coordenadas relativas.
PD[X, y][, z][[ x, y][, z ]... ]	Mueve la pluma actual a la posición solicitada, realizando el trazo desde el punto de actual.
PU[X, y][, z][[ x, y][, z ]... ]	Mueve la pluma actual a la posición solicitada, sin realizar el trazo.
OTROS COMANDOS	
SP[ 0, 1, 2... ]	Selecciona una pluma o una herramienta nueva para el uso. Si no se da ningún número de la pluma o un valor de cero, el controlador realiza un EOF (final del comando del archivo). Una vez que se realice un EOF, no se ejecuta ningún movimiento hasta que un nuevo comando IN se recibe.

La lista de comandos es grande, en realidad no todos los comandos son usados, esto depende del modelo del trazador y del software que genera la información. Para nuestra mesa trazadora sólo se manejan los comandos antes mencionados con sus respectivas coordenadas X – Y. El listado siguiente muestra el inicio y final del archivo ASCII que contiene la información del gráfico de la Figura 2.9.

**IN;PU6910,555;SP1;PD6910,555;PD6915,555;PD6920,555;PD6925,555;PD6930,555;PD6935,555;PD6940,555;PD6945,560;PD6950,560;PD6955,565;PD6960,565;PD6960,570;PD6965,570;PD6970,575;PD6975,580;PD6975,585;PD6980,585;PD6980,590;PD6980,595;PD6985,600;PD6985,605;PD6985,610;PD6985,615;PD6985,620;PD6985,625;PD6985,630;PD6985,635;PD698.....**

.....  
.....  
**PD3985,5455;PD3995,5450;PD4000,5450;PD4010,5450;PD4020,5445;PD4030,5445;PD4040,5445;PD4050,5445;PD4060,5445;PD4070,5445;PD4080,5450;PD4090,5450;PD4100,5455;PD4105,5455;PD4115,5460;PD4125,5465;PD4135,5470;PD4140,5475;PD4150,5480;PD4160,5485;PD4165,5490;PD4175,5495;PD4180,5505;PD4185,5510;PD4195,5520;PD4200,5525;PD4205,5535;PD4210,5540;PD4215,5550;PD4220,5560;PD4220,5570;PD4225,5575;PD4230,5585;PD4230,5590;PU3880,5535;PD3825,5515;PU0,0;SP0;PG;**

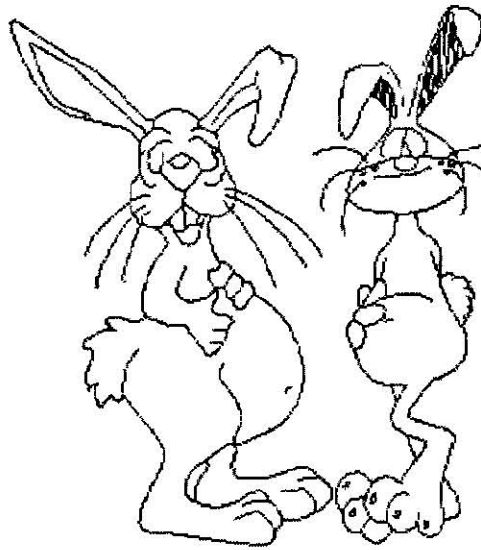
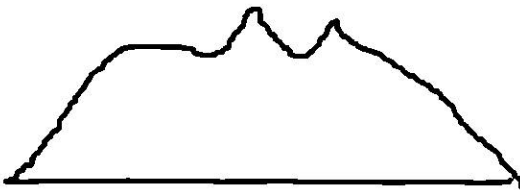


Figura 2.9 Imagen de un archivo HPGL.

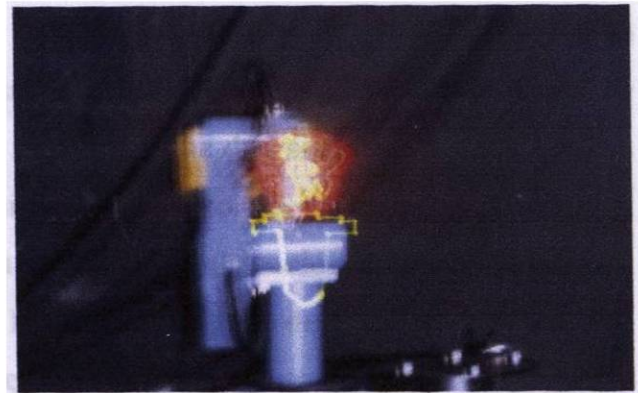
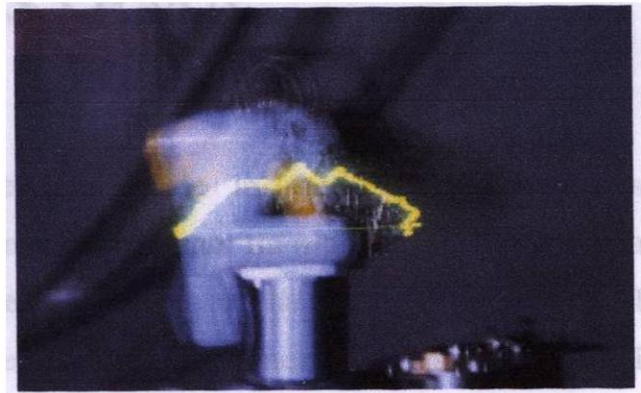
En el listado anterior la coma se usa como separador y el punto y coma como terminador, los comandos PU (pluma arriba) y PD (pluma a bajo) tienen a continuación las coordenadas absolutas X – Y, a donde debe dirigirse la pluma o herramienta de corte, los comandos SP1, SP0 (selección de pluma), IN (inicialización), PG (alimentación de página) son ignorados sin afectar los trazos. Los valores de las coordenadas están dados en unidades de trazador (plu), donde, 1,016 plu = 1 pulgada. Las aplicaciones tipo CAD como: ORCAD, AUTOCAD, COREL DRAW, etc., generan archivos de trazador (\*.PLT) con una mayor variedad de comandos HPGL.

En seguida se muestra una aplicación de cómo los archivos HPGL pueden ser utilizados para realizar trayectorias (Figura 2.10).

## IMAGENES



## FOTOGRAFIAS



**Figura 2.10 Brazo manipulador siguiendo las trayectorias indicadas por el archivo HPGL.**

# CAPÍTULO 3

## MODELADO DEL SISTEMA.

### 3.1 Introducción.

En este capítulo se muestra los elementos de la mesa de trazos y sus características. Se realiza un análisis del sistema para obtener un modelo cinemático para controlar la posición y velocidad de la trayectoria de la luz láser. No se considera el modelo dinámico ya que el análisis de las fuerzas que intervienen en el sistema no es el objetivo para el proyecto.

Los motores, cuya velocidad y ángulo de giro serán regulados por los pulsos de entrada del controlador del motor, no manejan directamente los rieles de desplazamiento sino que lo hacen a través de un sistema reductor conformados por engranes. Por lo tanto, es necesario considerar esta etapa en el modelado. En esta etapa existe una relación entre la cantidad de pasos dados por el motor y la distancia recorrida por el riel de desplazamiento. Esto de acuerdo al radio o al número de dientes de los engranes. Por lo tanto la representación del sistema de la mesa de trazo comprende los dos motores de pasos y los dos sistemas de engranes. Ambos motores y los dos sistemas reductores son idénticos. Cada motor con su respectivo sistema reductor se puede considerar en conjunto como un solo sistema “motor – reductor”. La combinación de ambos sistemas “motor – reductor” en la mesa de trazos, hacen posible el desplazamiento en dos dimensiones, cada sistema se asigna a cada riel de desplazamiento formando un sistema de desplazamiento en coordenadas cartesianas  $X \times Y$ .

### 3.2 Datos de los componentes eléctricos y mecánicos.

Dentro de los componentes eléctricos se consideran a los motores de pasos, ya que estos son controlados por medio de corrientes eléctricas que alimentan cada una de sus bobinas. Los motores utilizados son motores de pasos de imán permanente.



Los componentes mecánicos son los engranes que conforman los sistemas reductores así como la cremallera que convierte el movimiento angular del sistema de engranes en un movimiento lineal.

No se toman en cuenta los datos del láser ya que el objetivo es el de generar las trayectorias de corte y no en sí el corte con láser.

### 3.2.1 Especificaciones de los motores.

Como se mencionó anteriormente los motores de paso son idénticos en características y poseen los siguientes valores nominales obtenidos de sus datos de placa.

- Voltaje: 5 volts C. D.
- Corriente: 1 ampere C. D.
- Potencia: 5 Watts
- Ángulo de paso ( $\theta_s$ ):  $1.8^\circ$

Cada motor está empotrado en una base desde donde se conecta al sistema reductor de cada eje de desplazamiento.

### 3.2.2 Especificaciones del sistema reductor.

El sistema reductor está compuesto por tres engranes y una cremallera como se muestra en la Figura 3.1.  $E_1$  es movido directamente por el motor a pasos, debido a esto ~~una cantidad de~~ <sup>ángulo</sup> grados que gira  $E_1$  es igual a la ~~cantidad de~~ <sup>ángulo</sup> grados que gira el motor de pasos.  $E_1$  se conecta por medio de sus dientes a  $E_2$  y este está unido a  $E_3$  por medio de un eje el cual los conecta desde su centro. Este eje es perpendicular a la rotación de ambos engranes ( $E_2$  y  $E_3$ ). El  $E_3$  se desplaza a través de la cremallera convirtiendo el movimiento angular de los engranes en un movimiento lineal. Los datos de cada engrane y de la cremallera se pueden observar en la Figura 3.2. La Figura muestra el número de dientes, el diámetro y el perímetro de cada engrane.

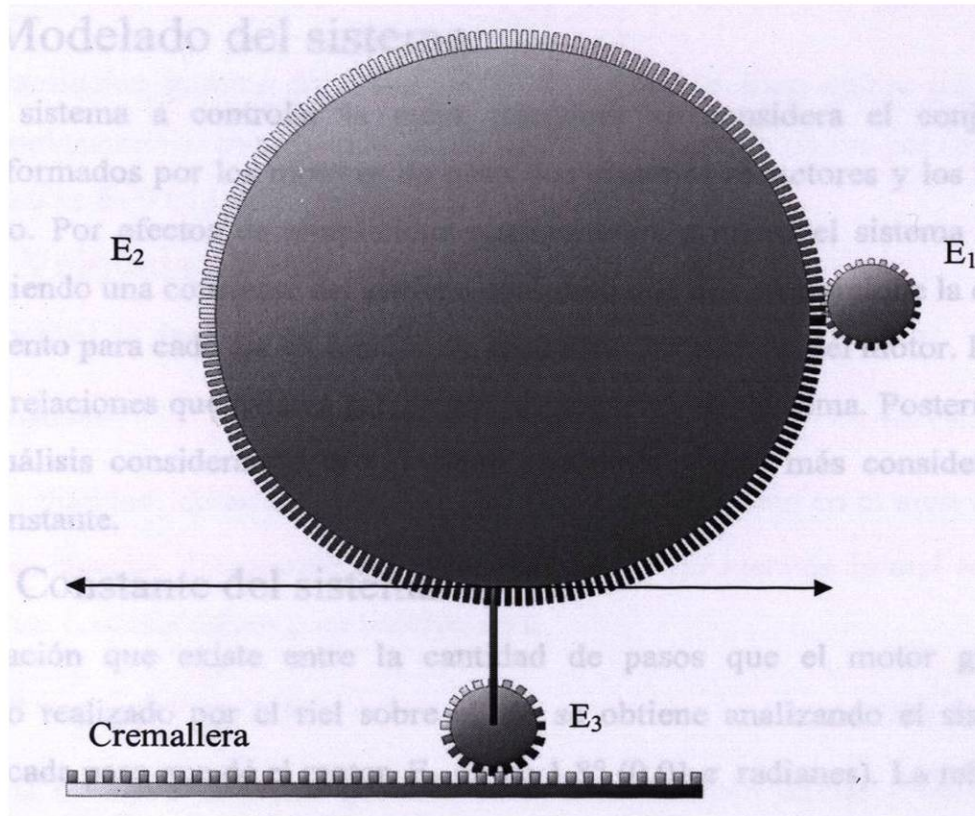


Figura 3.1 Sistema reductor.

	E <sub>1</sub>	E <sub>2</sub>	E <sub>3</sub>
Número de dientes (N)	N <sub>1</sub> = 24	N <sub>2</sub> = 185	N <sub>3</sub> = 20
Diámetro (D)	D <sub>1</sub> = 0.835 cm.	D <sub>2</sub> = 5.925 cm.	D <sub>3</sub> = 0.6403997 cm.
Perímetro (P)	P <sub>1</sub> = 2.6232298 cm.	P <sub>2</sub> = 18.6139364 cm	P <sub>3</sub> = 2.011875 cm

	Cremallera
Número de dientes/cm.	9.94

Figura 3.2 Dimensiones y características de los engranes.

Tabla

### 3.3 Modelado del sistema.

Como sistema a controlar la mesa trazadora se considera el conjunto de componentes formados por los motores de paso, los sistemas reductores y los rieles de desplazamiento. Por efectos de simplicidad analizaremos primero el sistema motor – reductor obteniendo una constante del sistema completo que nos proporcione la distancia de desplazamiento para cada eje en función de los pulsos de entrada del motor. Para esto se utilizan las relaciones que existen entre cada componente del sistema. Posteriormente se hace un análisis considerando la velocidad constante y uno más considerando la aceleración constante.

#### 3.3.1 Constante del sistema.

La relación que existe entre la cantidad de pasos que el motor gira y el desplazamiento realizado por el riel sobre el eje se obtiene analizando el sistema de engranes. Por cada paso que dé el motor,  $E_1$  girara  $1.8^\circ$  ( $0.01\pi$  radianes). La relación de número dientes entre  $E_1$  y  $E_2$  es igual a

$$N = \frac{N_1}{N_2}. \quad \text{ec. (3.1)}$$

Entonces  $E_2$  gira un ángulo de  $\frac{N_1}{N_2}(0.01\pi)$  radianes por cada paso. Como  $E_3$  gira en el mismo eje que  $E_2$  entonces el ángulo de giro de  $E_2$  y de  $E_3$  son idénticos. Además sabemos que:

$$\theta_3 = \frac{S_3}{r_3}, \quad \text{ec. (3.2)}$$

despejando  $S_3$ , se obtiene la  $S_3 = \theta_3 r_3$ , como  $D_3 = 2r_3$  se tiene que  $S_3 = \frac{\theta_3 D_3}{2}$  y con esto se obtiene el desplazamiento lineal. Sustituyendo ecuaciones se tiene que la constante del sistema es:

$$K = S_3 = \frac{N_1}{N_2}(0.01\pi)\frac{D_3}{2} = \frac{N_1}{N_2}(0.005\pi)D_3, \quad \text{ec. (3.3)}$$

$$K = \frac{24}{185} \frac{(0.005\pi)}{\text{paso}} \times 0.6403997 \text{ cm} = 0.01305 \frac{\text{mm}}{\text{paso}}, \quad \text{ec. (3.4)}$$

la cual es la resolución mínima del sistema motor – reductor. Esto quiere decir que la cantidad de desplazamiento mínimo que puede realizar la mesa de trazos por cada eje en forma individual es de 0.013 mm aproximadamente.

### 3.3.2 Velocidad constante.

La velocidad de desplazamiento del sistema está dada por la ecuación

$$V = k \frac{dp}{dt} = \text{constante}, \quad \text{ec. (3.5)}$$

donde  $V$  es la velocidad,  $dp$  es el número de pulsos,  $dt$  es el cambio en el tiempo y  $k$  es la constante del sistema. Integrando esta expresión entre un instante inicial cero y un instante posterior  $t$ , obtendremos  $p$  en función de  $t$ :

$$dp = \frac{V}{k} dt \Rightarrow p = \frac{V}{k} \int dt = \frac{V}{k} t + p_0, \quad \text{ec. (3.6)}$$

donde  $p_0$  es la cantidad de pasos que existe entre la posición inicial y un punto de referencia.

### 3.3.3 Aceleración constante.

Si suponemos que el desplazamiento es con una aceleración constante según la ecuación

$$a = \frac{dV}{dt} = \text{constante}, \quad \text{ec. (3.7)}$$

donde  $a$  es la aceleración,  $dV$  es el cambio de velocidad y  $dt$  es el cambio en el tiempo. Podemos obtener la aceleración en función del tiempo integrando esta expresión desde un instante inicial cero a un instante posterior  $t$

$$dV = a dt \Rightarrow V = a \int dt = at + V_0, \quad \text{ec. (3.8)}$$

donde  $V_0$  representa la velocidad inicial del movimiento. Ahora sustituyendo esta última ecuación en Ecuación 3.6 se obtiene

$$dp = \frac{(at + V_0)}{k} dt \Rightarrow p = \int \left( \frac{at + V_0}{k} \right) dt = \frac{1}{2} \frac{at^2}{k} + \frac{V_0}{k} t + p_0. \quad \text{ec. (3.9)}$$

Eliminando el tiempo entre la Ecuación 3.8 y la Ecuación 3.9 se puede ver que

$$V^2 = V_0^2 + 2ak(p - p_0) \quad \text{ec. (3.10)}$$

Con esto se obtiene la velocidad en función de la posición en un movimiento con aceleración constante.

Con todo lo anterior, es posible determinar las velocidades y las aceleraciones a través de una trayectoria seguida por el haz del láser.

# CAPITULO 4

## DESARROLLO E IMPLEMENTACIÓN DEL PROYECTO.

### 4.1 Introducción.

Para poder realizar el control de las trayectorias de la luz láser es necesario tomar en cuenta ciertas condiciones. El sistema de control debe de permitir el movimiento de cada eje en forma independiente y la frecuencia de los pulsos de entrada del controlador del motor debe de permitir el giro del motor sin la pérdida de pasos. Con estas características se logra que las trayectorias generadas por la mesa de trazos tengan un error menor con respecto a las trayectorias reales. De acuerdo a estos criterios considerados para determinar el modo de operación, el funcionamiento del equipo y la implementación, se realiza el análisis y diseño en tres secciones:

- La mesa de trazos.
- El controlador de los motores.
- Sistema generador de trayectorias (Computadora).

En el capítulo se retoma el análisis cinemático desarrollado en el capítulo anterior para la mesa de trazos. Posteriormente se describe el diseño del controlador de motores a paso, el cual debe de cumplir con las características antes mencionadas, y se implementa en una GAL. Por último, se explica como la computadora envía los datos al controlador de los motores de pasos.

### 4.2 Diagrama general.

En la Figura 4.1 se puede observar el diagrama a bloques del proyecto. En esta se aprecia cada uno de los componentes del proyecto en una forma general.

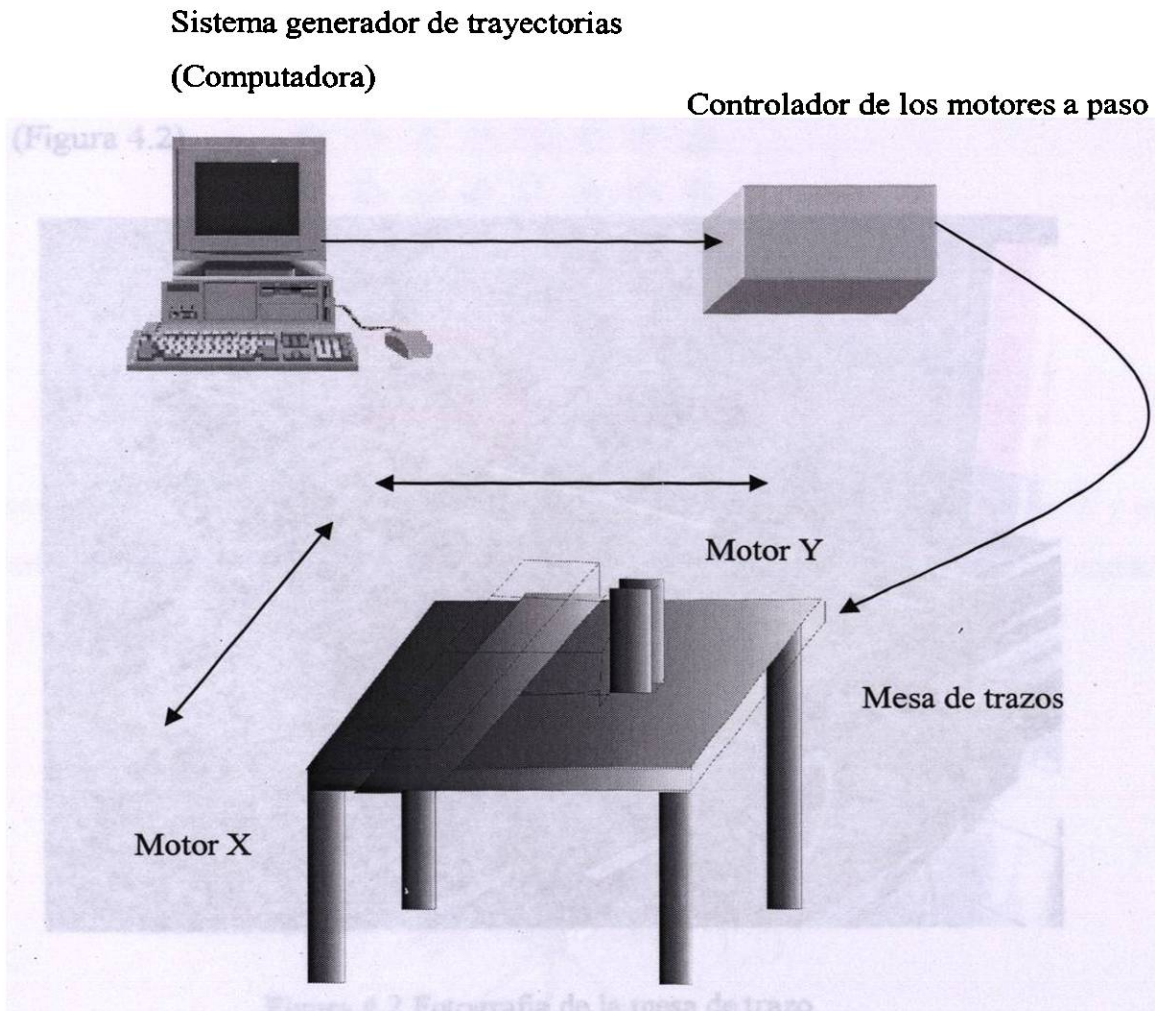


Figura 4.1 Diagrama general de la mesa de trazo.

En la Figura 4.1 se observan los componentes mencionados en la introducción. El sistema generador de trayectorias (la computadora) envía los datos necesarios para la generación de la trayectoria hacia el controlador de los motores de paso. El controlador de los motores de pasos recibe los datos y los interpreta para convertirlos en una rotación de los motores. La mesa de trazos convierte el movimiento rotacional de los motores en la trayectoria que debe seguir el haz de luz láser. Como se puede ver en el diagrama este esquema de control es en lazo abierto ya que no hay una retroalimentación de la posición en la que se encuentra el haz de luz hacia el generador de trayectorias.

Debido a las cualidades de los motores a paso y a las velocidades que se maneja, no es necesario realizar un esquema de control de lazo cerrado.

### 4.3 Mesa de trazos.

La mesa de trazos consiste en una mesa rectangular que cuenta con dos sistemas de rieles que realizan el desplazamiento del haz de luz láser a través de todo el plano de la mesa (Figura 4.2).

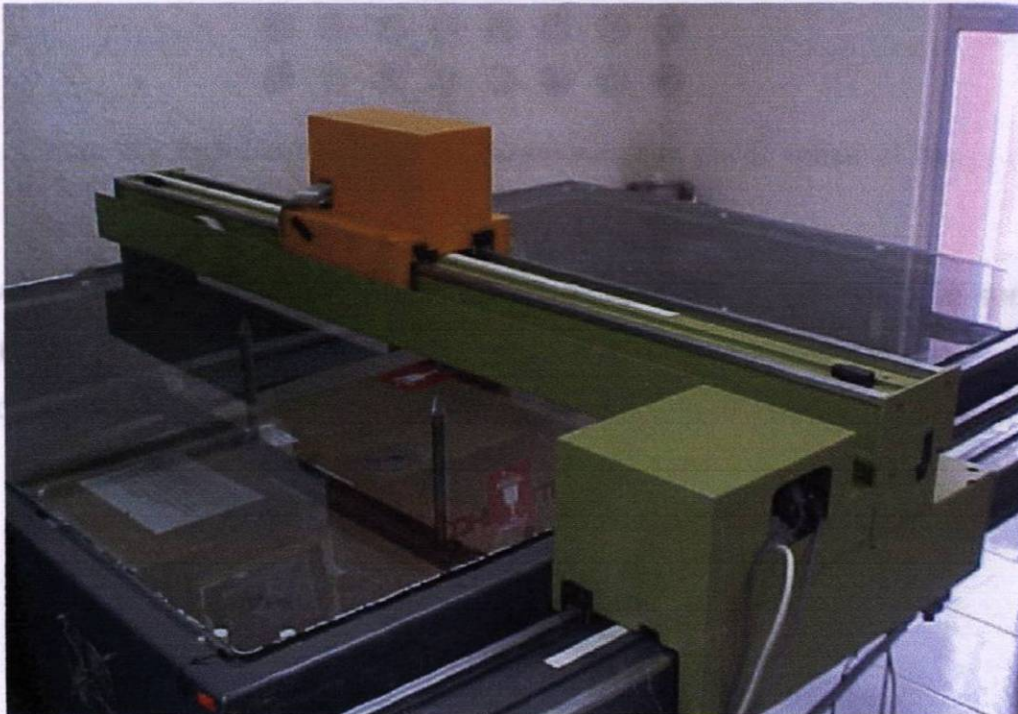


Figura 4.2 Fotografía de la mesa de trazo.

Cada sistema de rieles es movido a través de un sistema de engranes motor – reductor. Ya que los motores de paso solo pueden moverse ángulos discretos, la mesa de trazo solo puede moverse distancias discretas. El sistema de engranes antes mencionado transforma cada paso angular que da un motor, en un paso lineal que equivale aproximadamente a 0.013 mm, que es la resolución de la mesa. Las posiciones que puede tomar el haz de luz láser pueden ser representadas por una matriz de puntos como se muestra en la Figura 4.3. Es decir, cualquier trayectoria o curva que se quiera realizar con el haz láser en la mesa de trazo tiene que ser ajustada a estos puntos.



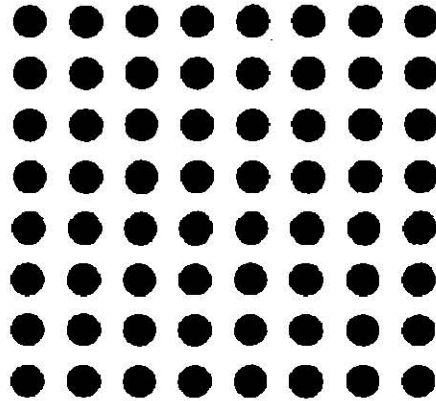


Figura 4.3 Representación de las posiciones que puede tomar el haz de luz láser.

Los movimientos posibles que pueden ser realizados desde una posición son 8 y están representadas en la Figura 4.4. El controlador de los motores debe de ser capaz de realizar cualquiera de estos movimientos.

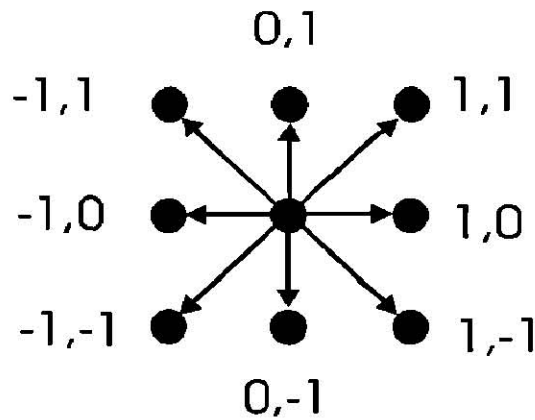


Figura 4.4. Desplazamientos posibles.

## 4.4 Controlador de los motores a pasos.

En este punto se muestra el diseño del controlador de los motores a pasos. Por efecto de simplicidad se hizo primeramente el diseño del controlador de un motor a pasos y posteriormente se implementó para tres motores a pasos (el tercero será para la implementación de un eje Z en el futuro).

Como se mencionó en el capítulo 2, los motores necesitan la inversión de la corriente que circula en sus bobinas en una secuencia determinada. Cada inversión de la polaridad provoca el movimiento del eje en un paso, cuyo sentido de giro está determinado por la secuencia seguida.

A continuación se puede ver la tabla con las secuencias necesarias para controlar motores de paso del tipo Bipolares (Figura 4.5):

Sentido 1

Paso	Terminales			
	A	B	C	D
1	1	0	1	0
2	1	0	0	1
3	0	1	0	1
4	0	1	1	0

Sentido 2

Paso	Terminales			
	A	B	C	D
1	1	0	1	0
2	1	0	0	1
3	0	1	0	1
4	0	1	1	0

Figura 4.5 Secuencias de los motores de paso.

Estas dos secuencias pueden ser representadas como un sistema secuencial síncrono. Para este tipo de sistemas, en el lenguaje ABEL-HDL, se puede utilizar las ecuaciones de los Flip Flops, la tabla de estados o la descripción del problema.

En la Figura 4.6 se muestra el diagrama de transición del sistema secuencial que representa al controlador del motor a pasos.

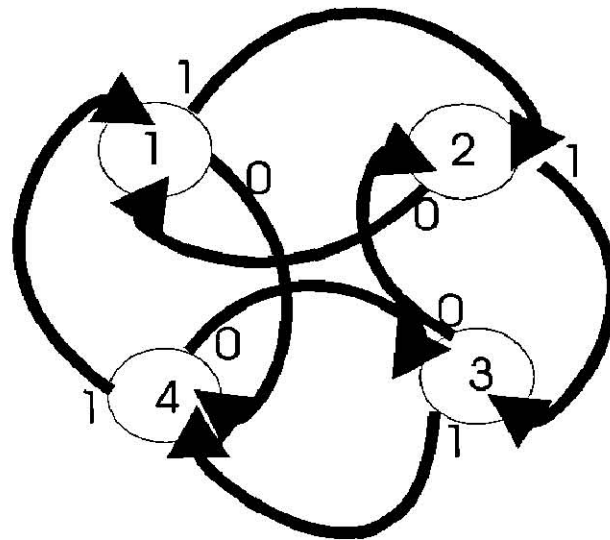


Figura 4.6 Diagrama de transición del controlador del motor a pasos.

Cada estado de este sistema secuencial representa un conjunto de valores para las terminales A, B, C y D del controlador, determinadas por los valores de las tablas de la Figura 4.5. Las transiciones entre los estados dependen de una entrada, S, que determina el sentido del giro del motor. Para S=1, las transiciones siguen las flechas en el sentido de las manecillas del reloj y para S=0, en contra de las manecillas del reloj.

En el diagrama de bloques (Figura 4.7), se tienen como entradas a S, que determina el sentido del giro del motor además de la señal de sincronía Clk, como salidas combinacionales, se encuentran las terminales A, B, C y D y como salidas secuenciales o registradas dos Flip Flops llamados Q1 y Q0 que sus combinaciones representa a cada uno de los cuatro posibles estados.

En la tabla de estados (Figura 4.8) se puede observar el estado próximo en función de la posible entrada, esta tabla se utiliza para la creación del archivo ABEL – HDL.

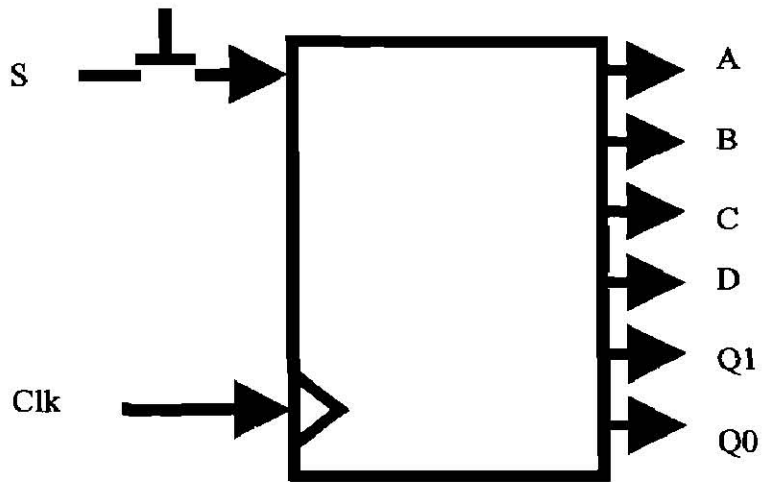


Figura 4.7 Diagrama a Bloques del controlador del motor a pasos.

	S=0	S=1
1	4	1
2	3	2
3	2	3
4	1	4

Figura 4.8 Tabla de estados del controlador del motor a pasos.

La asignación de valores a los estados se hace como sigue:

	Q1	Q0
1	0	0
2	0	1
3	1	0
4	1	1

Figura 4.9 Asignación de valores a los estados del controlador del motor a pasos.

La implementación de este sistema secuencial se realiza en un GAL16V8 y el archivo en formato ABEL – HDL de este sistema secuencial se presenta en el anexo.

Con esto se obtiene el controlador en un circuito integrado, pero con corrientes en las terminales de salida muy bajas. Para esto es necesario agregar una etapa de potencia entre las salidas del GAL16V8 y las entradas del motor de pasos. La forma de implementar esto es con transistores (sección 2.1.1). En la Figura 4.10 se muestra el circuito del controlador de un motor a pasos

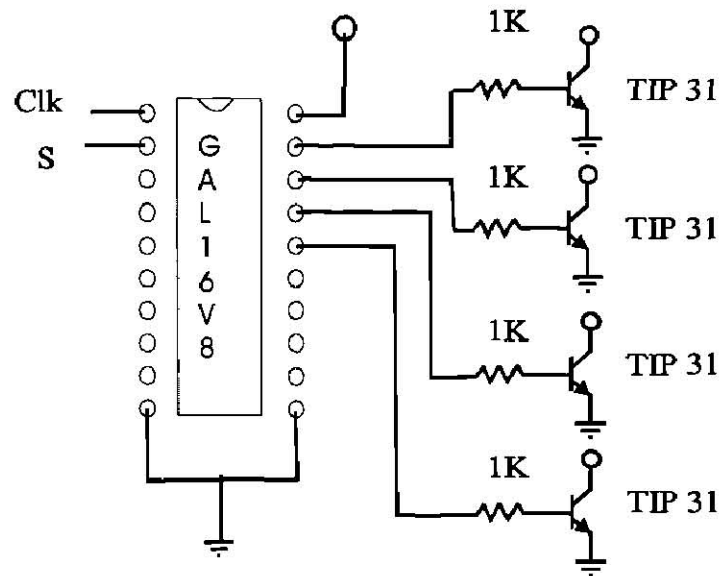


Figura 4.10 Circuito del controlador de un motor a pasos.

Para poder realizar el control de tres motores a la vez en forma independiente es necesario colocar en paralelo tres controladores como el diseñado anteriormente. En la Figura 4.11 se muestra el diagrama a bloques del controlador para 3 motores de paso. En esta se observa que el controlador tiene 6 variables de entradas, Clk x, Clk y, Clk z que son los pulsos de entrada de cada motor y Sx, Sy, Sz que son los sentidos de cada motor X, Y, Z respectivamente.

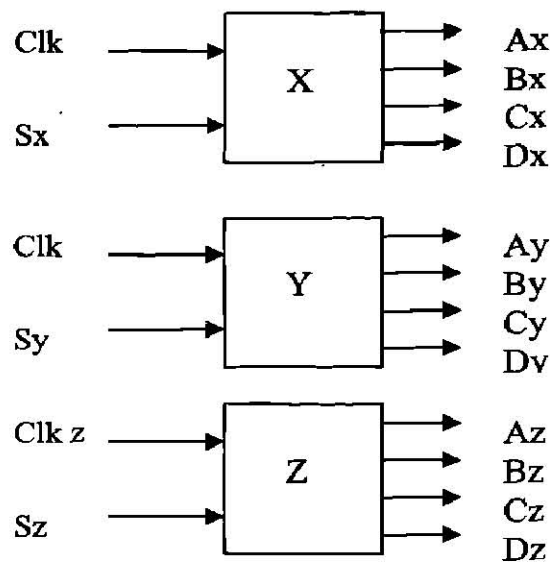


Figura 4.11 Diagrama a bloques de un controlador para 3 motores de paso.

## 4.5 Sistema generador de trayectorias (Computadora).

Para realizar cualquier movimiento es necesario enviar al controlador de los motores de paso la información necesaria para que lo realice. Para esto se utiliza una computadora, que además de realizar los cálculos de los puntos que sigue la trayectoria, envía a través del puerto paralelo la información necesaria para el movimiento. Esto lo hace utilizando 6 bits de información.

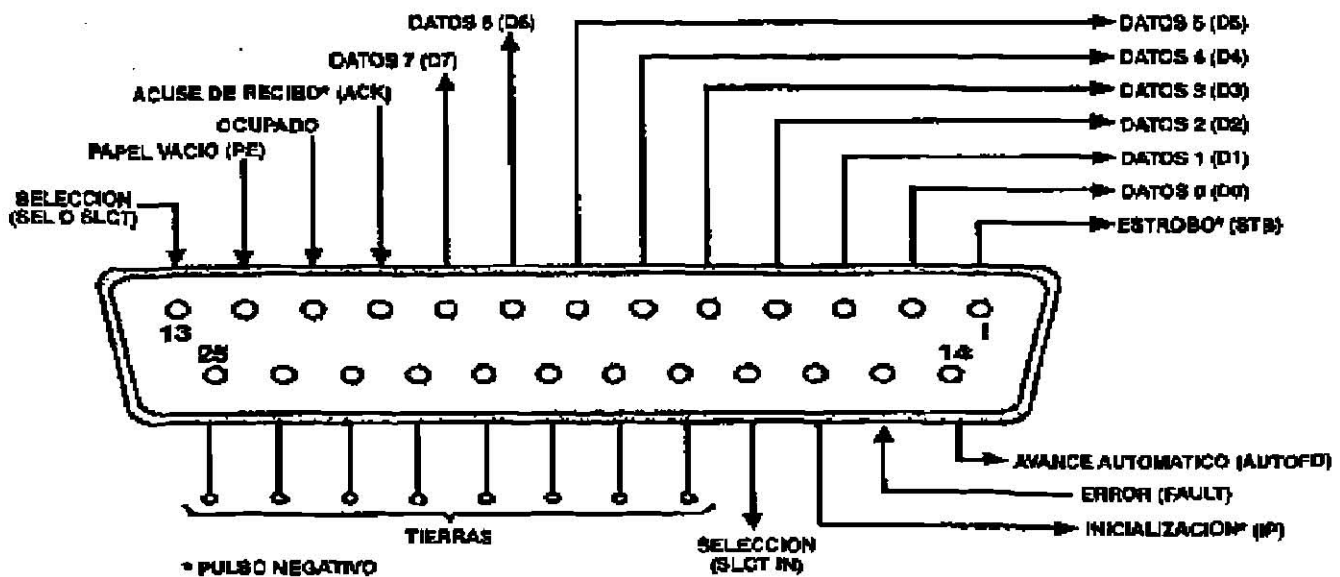


Figura 4.12 Puerto paralelo (Interfase DB25).

La información que requiere el controlador es la cantidad de pasos que debe girar cada motor y el sentido de giro de cada uno de los motores. Esta información es enviada a través del puerto paralelo (Figura 4.12). En la Figura 4.13 se muestra una tabla con el pin del puerto utilizado y la información que representa.

Información	Pin del puerto
Clk x	D0
Sx	D1
Clk y	D2
Sy	D3
Clk z	D4
Sz	D5
Corte	D6

Figura 4.13. Tabla de Datos utilizados en el puerto paralelo.

En la tabla anterior además de la información del controlador se tiene un dato más para activar el corte o desactivarlo. Lo anterior es para poder realizar trayectorias sin necesidad realizar el corte, esto desactivando el láser. En el momento que se desee cortar, se activa el láser nuevamente y se realiza la trayectoria.

En resumen los desplazamientos realizados por la mesa alcanzan una resolución de aproximadamente 0.013 mm. Cualquier punto dentro del plano de la trayectoria que se desea seguir, debe de ajustarse a un número múltiplo de 0.013 mm. Cada motor funciona en forma independiente, es decir, se puede mover uno o más motores a la vez sin que esto afecte entre sí. El controlador de los motores necesita datos en forma de bits para poder realizar el control de los motores de paso.

# CAPÍTULO 5

## DISEÑO DE LA INTERFASE.

### 5.1 Introducción.

El maquinado de curvas planas es muy común en el (CNC). Usualmente se lleva a cabo por aproximaciones de estas curvas, con segmentos de líneas rectas o arcos circulares, a través de las interpolaciones realizadas por el CNC. En el desarrollo del proyecto se utilizan interpolaciones lineales ya que los puntos obtenidos son muy cercanos uno del otro. Esto evita un menor margen de error en el trazo de la figura.

En este capítulo se define el punto y la forma de representarse. Posteriormente, se muestra un algoritmo para generar trazos de líneas rectas, las cuales son la parte medular del trazo de las figuras. Enseguida, se muestra un algoritmo para la generación de círculos, este toma en cuenta la resolución de la mesa para tener el mínimo de error para el trazo de la trayectoria. Por último, se muestra como realizar trazos de figuras en lenguaje HPGL.

parámetro  
?

### 5.2 Trazos de figuras.

#### 5.2.1 Punto.

La mesa, como se mencionó en el capítulo 4, puede posicionar el haz de luz láser en posiciones discretas de múltiplos de 0.013 mm que equivalen a un paso del motor. Por cada paso de un motor podemos definir un punto en "x" o en "y" según sea el desplazamiento.

El sistema de desplazamiento se puede referir de dos formas diferentes:

- a) Coordenadas absolutas: posición fija de la mesa como punto de referencia.
- b) Coordenadas relativas: posición actual del haz como punto de referencia para el sistema de coordenadas.

En el caso de las coordenadas absolutas, la mesa se toma como un sistema coordenado XY, con un origen fijo en  $x_0=0$ ,  $y_0=0$  en un punto de la mesa. Este punto se encuentra ilustrado en la Figura 5.1.



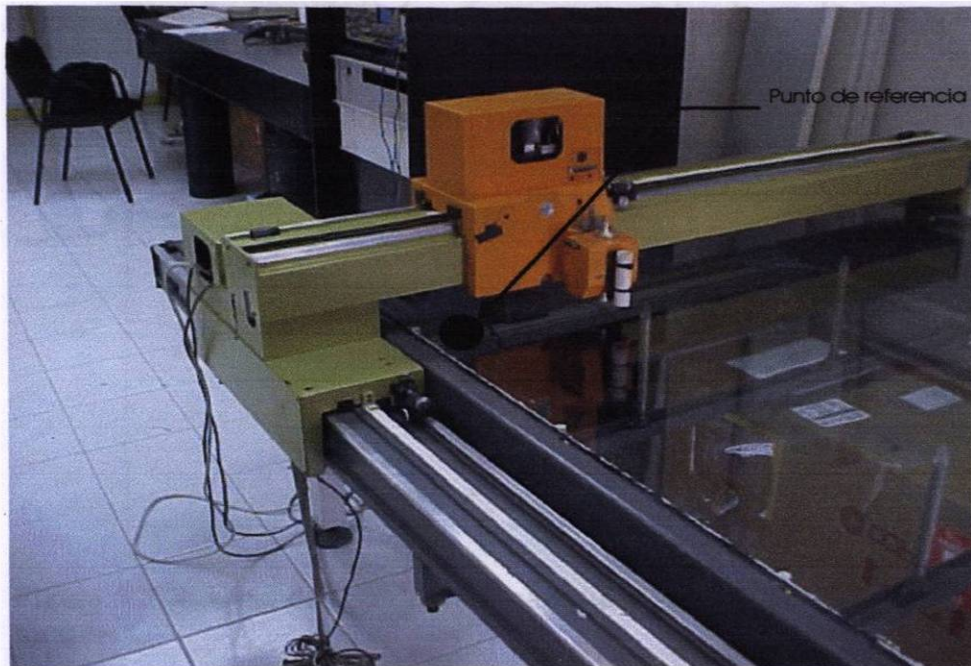


Figura 5.1 Localización del punto de referencia del sistema de coordenadas cartesianas absolutas manejadas por la mesa trazadora.

Se puede representar una coordenada  $(x_n, y_n)$  como el número de pasos que existen del eje “X” y del eje “Y” multiplicado por la resolución de la mesa trazadora. Esto es:

$$x_n = (\# \text{pasosX})(\text{resolución}) \text{ y} \quad \text{ec. (5.1)}$$

$$y_n = (\# \text{pasosY})(\text{resolución}), \quad \text{ec. (5.2)}$$

donde “# pasosX” y “# pasosY” es la cantidad de pasos que existe entre el punto de referencia  $(x_0, y_0)$  y el punto  $(x_n, y_n)$ , en el eje “X” y en el eje “Y” respectivamente. En la Figura 5.2 se representa lo anterior.

En el caso de las coordenadas relativas, se toma como punto de referencia  $(x'_0, y'_0)$  el punto en donde se encuentra localizado el haz,  $x'_n$  y  $y'_n$  es la cantidad de pasos que existe en el eje “X” y en el eje “Y” respectivamente, desde el punto  $(x'_0, y'_0)$ , que es el nuevo punto de referencia, hasta el punto  $(x'_n, y'_n)$ . Esto se muestra en la Figura 5.3.

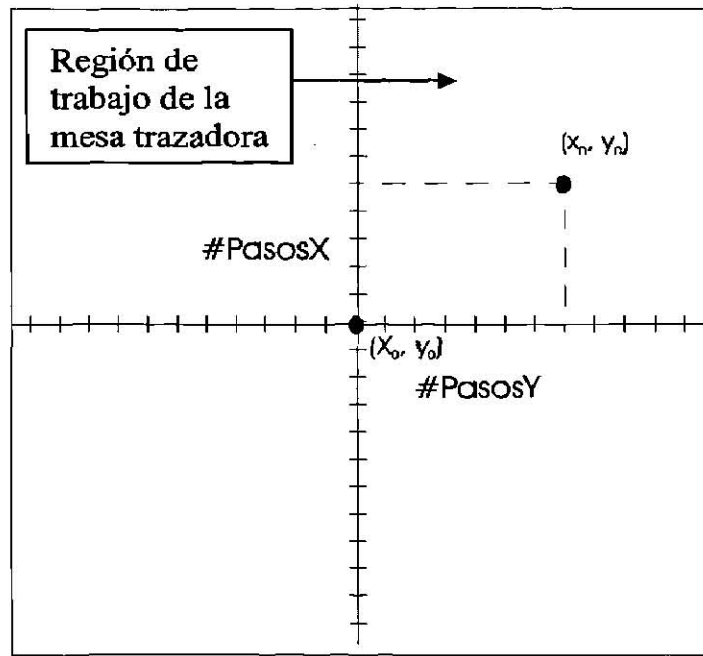


Figura 5.2 Sistema de coordenadas cartesianas absolutas de la mesa trazadora.

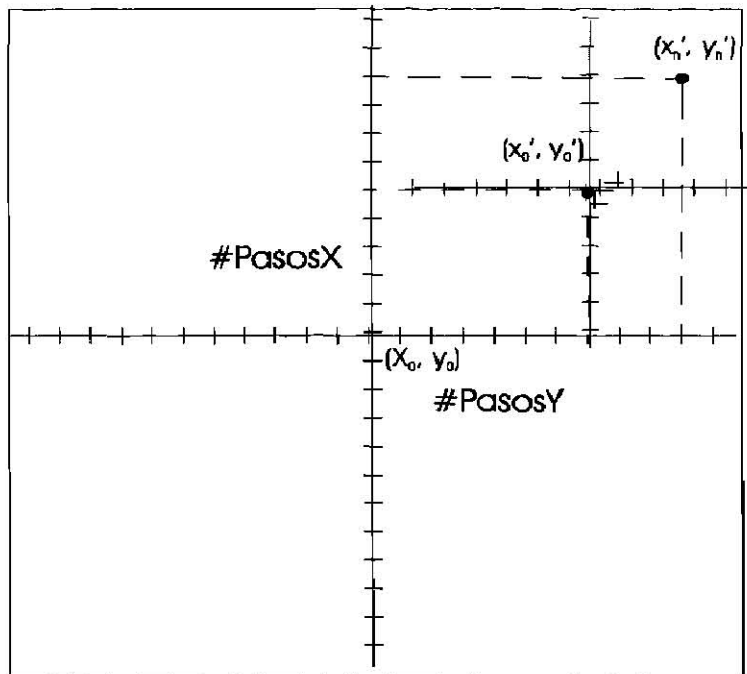


Figura 5.3 Sistema de coordenadas cartesianas relativas de la mesa trazadora.

### 5.2.2 Rectas.

Para definir una recta se necesitan dos puntos. Estos dos puntos son el punto inicial  $(x_0, y_0)$  y el punto final  $(x_n, y_n)$  de la recta. Con estos se puede formar un vector de movimiento  $\vec{V}$  que va desde el punto  $(x_0, y_0)$  hasta el punto  $(x_n, y_n)$ . Para poder realizar el trazo de la recta es necesario encontrar las componentes  $\vec{V}_x$  y  $\vec{V}_y$  del vector  $\vec{V}$  para definir la cantidad de pulsos que son necesarios enviar al controlador para cada motor. Esto es:

$$\vec{V}_x = (x_n - x_0)\hat{i} \text{ y} \quad \text{ec. (5.3)}$$

$$\vec{V}_y = (y_n - y_0)\hat{j}. \quad \text{ec. (5.4)}$$

La cantidad de pulsos que se deben de enviar al controlador son: para el motor "X",  $|\vec{V}_x|$  pulsos y para el motor "Y",  $|\vec{V}_y|$  pulsos. La dirección del motor X y del motor Y es el sentido de los vectores del movimiento. Esto se observa en la Figura 5.4.

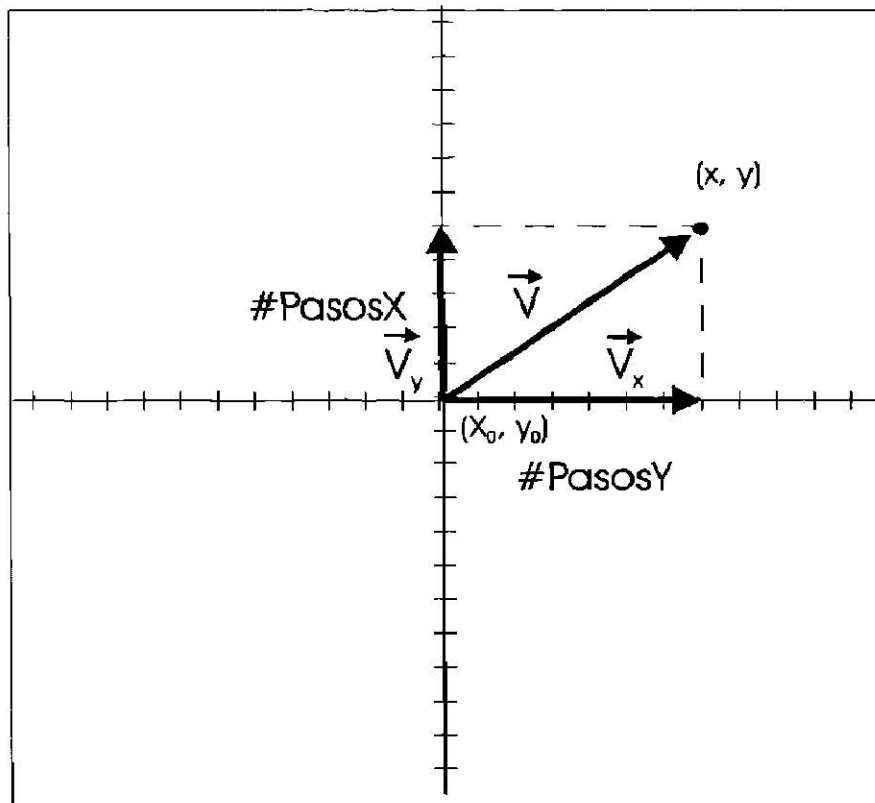


Figura 5.4 Vectores de desplazamiento para la trayectoria en línea recta.

En el algoritmo para el trazo de las trayectorias rectas se toman 3 casos generales de rectas:

- a) Las rectas con movimiento en eje "X"
- b) Las rectas con movimiento en eje "Y"
- c) Las rectas con movimiento en ambos ejes.

Las trayectorias de líneas rectas con movimiento en el eje "X" solo necesitan que el motor "X" gire la cantidad de pasos requerida. Para esto es necesario que se envíen al controlador el sentido y los pulsos necesarios para que gire el motor "X" hasta llegar al punto requerido. Lo mismo sucede para las trayectorias de líneas rectas con movimiento en el eje "Y". Para el movimiento en ambos ejes es necesario obtener la razón que existe entre los pasos de cada uno de los ejes por intervalo de tiempo. Con lo anterior se realiza el movimiento de los motores. Los pulsos enviados a los motores "X" y "Y" son representados en un diagrama de estados en la Figura 5.5.

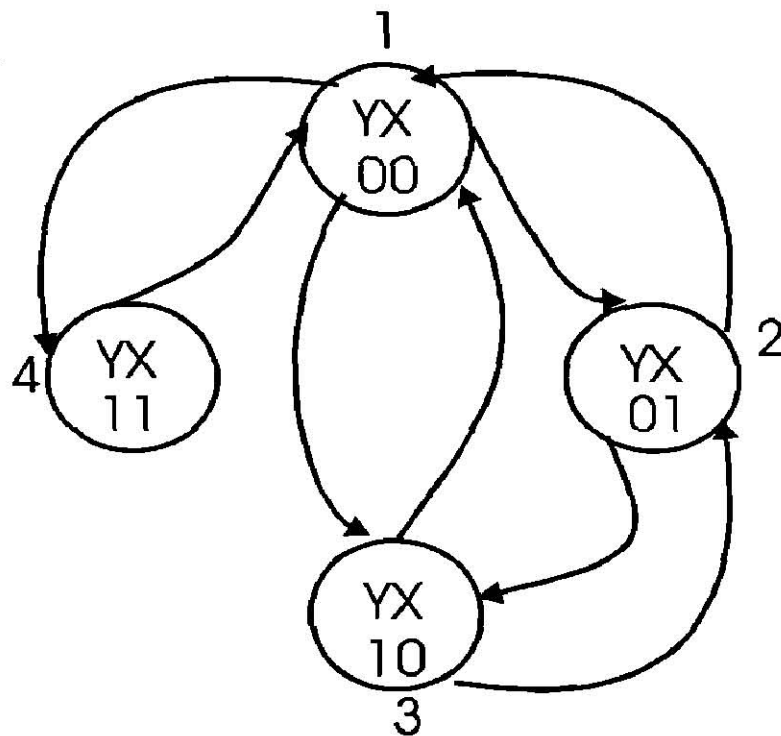


Figura 5.5 Diagrama de estados de la combinación de pulsos posibles recibidos por el controlador.

Para que el motor gire un paso, se le tiene que enviar una secuencia de un “1” seguido de un “0” al controlador del motor. Nunca se le debe de enviar dos “1” seguidos ya que siempre debe de regresar a “0”. Cuando se queda la señal en “0” el motor se mantiene estático. Una línea recta cualquiera se puede formar realizando combinaciones de los estados del diagrama de la Figura 5.5. Para encontrar la razón de cambio se encuentra la pendiente de la recta que se quiere trazar. Se presentan tres posibles casos:

- a) El valor absoluto de la pendiente de la recta es igual a uno
- b) El valor absoluto de la pendiente de la recta es menor que uno
- c) El valor absoluto de la pendiente de la recta es mayor que uno

Si el valor absoluto de la pendiente es igual a uno, es decir  $|m| = 1$ , los dos motores tienen que realizar el giro al mismo tiempo y a la misma velocidad. Para esto es necesario que el controlador de los motores reciba las entradas de los estados 1 y 4 en forma alternada.

Ejemplo:  $P_0 = (0,0)$ ,  $P_n = (3,3)$ , la recta formada por estos dos puntos tiene una

$|m| = \left| \frac{3-0}{3-0} \right| = 1$ , Por lo tanto la secuencia de estados enviada al controlador es: Estado 4,

Estado 1, Estado 4, Estado 1, Estado 4, Estado 1. El sentido de los motores “X” y “Y” se determinan con los signos de  $V_x$  y de  $V_y$  respectivamente. En este caso ambos vectores son positivos, por consecuencia el movimiento de ambos motores debe ser en sentido positivo y se le envían a los controladores de ambos motores una señal de “0”. Si el valor absoluto de la pendiente de la recta que se desea trazar es menor que 1, por cada intervalo de tiempo se manda un “0” o un “1” en forma alternada al motor “X”, es decir que por cada dos intervalos de tiempo, el motor “X” gira un paso. Al motor “Y” se le envía un pulso cada vez que  $y$  sea un entero en la siguiente ecuación

$$y = mx, \quad \text{ec. (5.5)}$$

donde  $y$  representa los pasos en el eje “Y”,  $x$  los pasos en el eje “X” y  $m$  la pendiente de la recta. Ejemplo:  $P_0 = (0,0)$ ,  $P_n = (4,-2)$ , la recta formada por estos puntos tiene una

$$|m| = \frac{1}{2}.$$

La secuencia de estados para el trazo de esta recta es: Estado 2, Estado1, Estado 4, Estado 1, Estado 2, Estado 1, Estado 4, Estado 1. El sentido del motor “X” es positivo ya que  $V_x = 4$  es positivo, el sentido del motor “Y” es negativo dado que  $V_y = -2$  es negativo.

Si el valor absoluto de la pendiente de la recta que se desea trazar es mayor que 1, por cada intervalo de tiempo se manda un “0” o un “1” en forma alternada al motor “Y”, es decir que por cada dos intervalos de tiempo, el motor “Y” gira un paso. Al motor “X” se le envía un pulso cada vez que  $x$  sea un entero en la siguiente ecuación

$$x = \frac{y}{m}, \quad \text{ec. (5.6)}$$

ejemplo:  $P_0 = (0,0)$ ,  $P_n = (-2,-5)$ , la recta formada por estos puntos tiene un  $|m| = \frac{5}{2}$ , la secuencia de estados para el trazo de esta recta sería: Estado 3 Estado 1, Estado 3, Estado 2, Estado 3, Estado 1, Estado 3, Estado 1, Estado 4, Estado 1. El sentido de ambos motores son negativos ya que  $V_x = -2$  y  $V_y = -5$ .

### 5.2.3 Círculos.

Para realizar el trazo de un círculo es necesario representarlo en ecuaciones paramétricas para poder obtener los puntos de este. Las coordenadas paramétricas del círculo son:

$$x = r \cos \theta \text{ y} \quad \text{ec. (5.7)}$$

$$y = r \sen \theta \quad \text{ec. (5.8)}$$

donde  $r$  es el radio del círculo y el ángulo  $\theta$  es el parámetro que se varía para poder realizar las interpolaciones y representa el ángulo en el cual está posicionado el punto  $(x, y)$  con respecto al eje X como se muestra en la Figura 5.6.

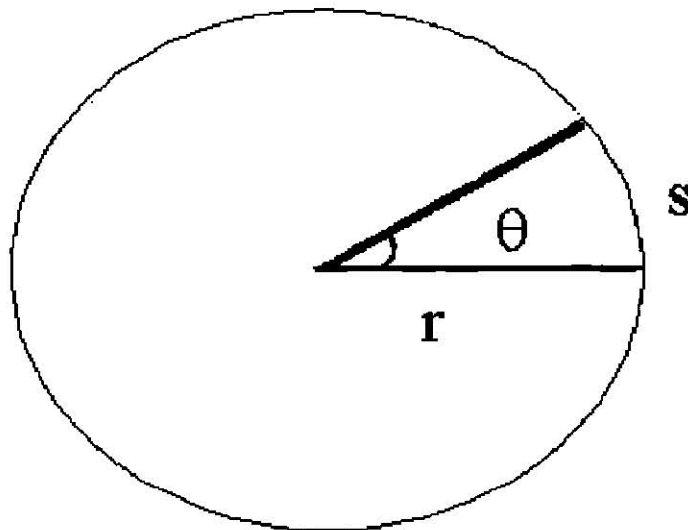


Figura 5.6 Componentes del círculo.

Para obtener las interpolaciones del círculo es necesario variar el ángulo  $\theta$  en las ec. (5.7) y ec. (5.8). El ángulo de variación  $\Delta\theta$  es el ángulo óptimo para incrementar el ángulo  $\theta$  sin perder resolución en el trazo del círculo. El  $\Delta\theta$  se obtiene con la ecuación siguiente:

$$\Delta\theta = \frac{0.013mm}{r}, \quad \text{ec. (5.9)}$$

donde  $\Delta\theta$  es el ángulo de variación, 0.013mm es la resolución de la mesa y  $r$  es el radio del círculo.

Los pasos que se siguen para el trazo de círculos son:

- 1.- Pedir el radio del círculo.
- 2.- Obtener el ángulo de variación  $\Delta\theta$  con la ec (5.9).
- 3.- Obtener el primer punto con las ec (5.7) y ec (5.8) y el ángulo inicial igual a cero.
- 4.- Obtener el punto siguiente incrementando el ángulo en  $\Delta\theta$ .
- 5.- Trazar una recta con el punto anterior como  $P_0$  y con el punto siguiente.  $P_n$ .
- 6.- Si el ángulo  $\theta$  es menor o igual que  $2\pi$  radianes pasar al punto 4, si no lo es terminar el proceso.

Se muestra el diagrama de flujo (Figura 5.7):

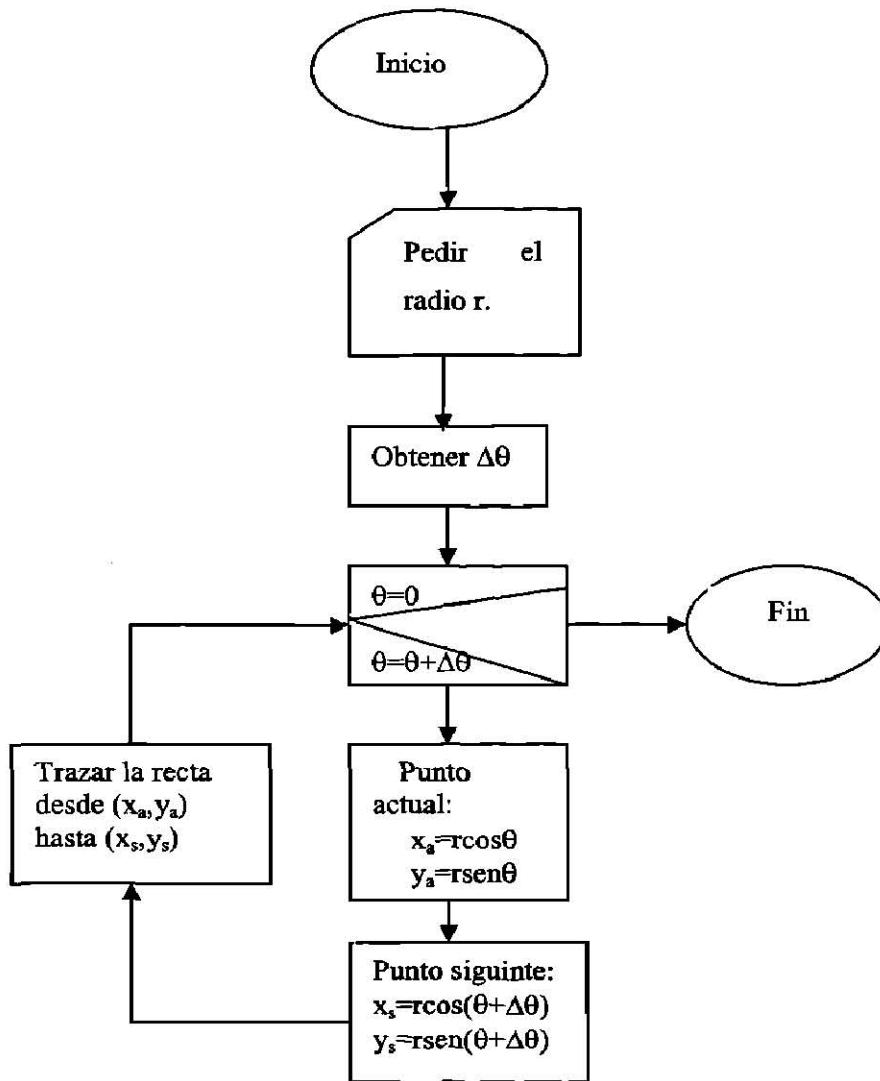


Figura 5.7 Diagrama de flujo para el trazo de un círculo.

### 5.2.4 Archivos HPGL.

La representación de una figura o un dibujo por medio del lenguaje HPGL es a través de puntos dentro de la trayectoria del dibujo. Para realizar el trazo de las trayectorias representadas con este tipo de lenguaje es necesario recorrer todos los puntos con el haz de luz láser. Para esta acción solo basta con unirlos por medio de líneas rectas. Por ejemplo: en la Figura 5.8 se representa un cuadro en lenguaje HPGL.



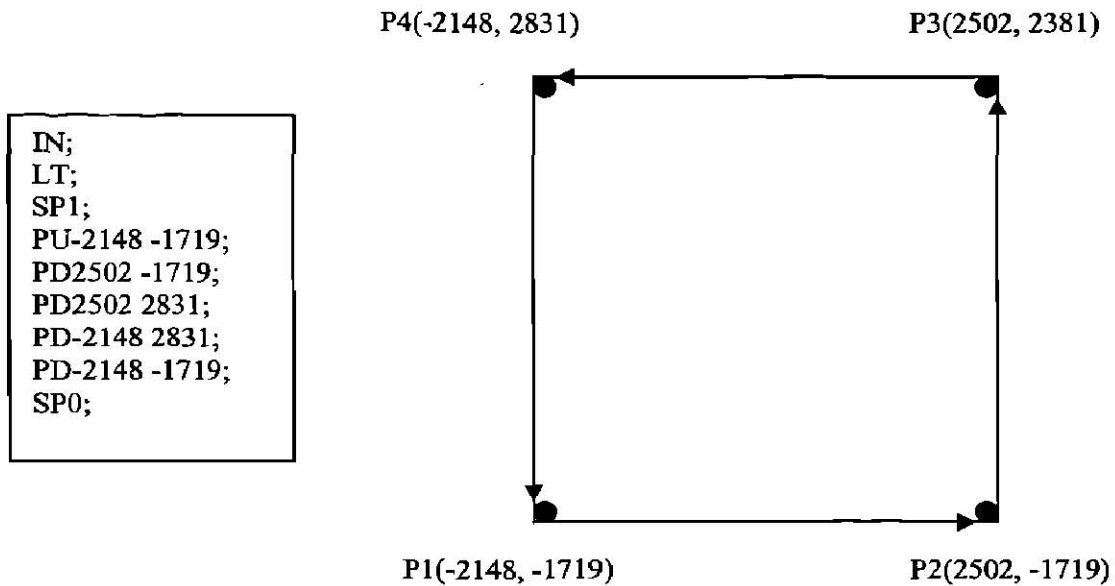


Figura 5.8 Archivo HPGL y trayectoria seguida por la herramienta.

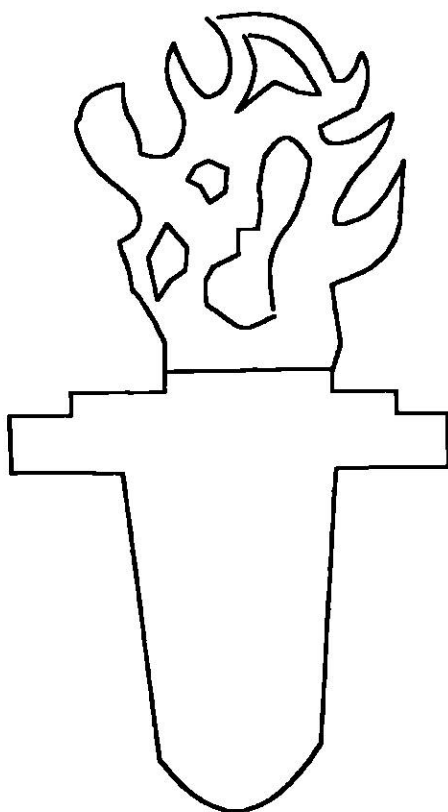
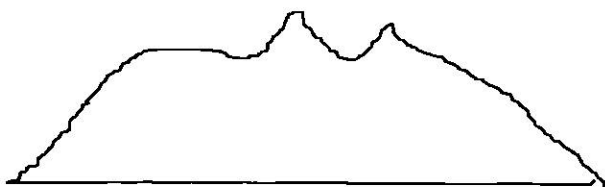
Aquí se observa (Figura 5.8) como, cada punto se une por medio de una línea recta. Cuando se tiene un punto con una línea de comando PU, solo se mueve el haz hacia ese punto sin realizar ningún corte durante la trayectoria recta y al presentarse un punto con una línea de comando PD, se mueve el haz al punto indicado realizando el corte durante la trayectoria recta. Encontrándose una línea de comando SP0 se termina la ejecución de la trayectoria. El resto de los comandos del HPGL no se toman en cuenta para este proyecto.

El algoritmo para poder realizar los cortes obteniendo los datos a través de un archivo HPGL es el siguiente:

- 1.- Se lee el comando. Si el comando es un punto, se nombra como posición siguiente y la posición donde se encuentra la herramienta se nombra como posición actual. Si el comando es SP0 se termina el trazo de la trayectoria.
- 2.- Si la posición siguiente tiene una línea de comando PU, se realiza la trayectoria en línea recta desde la posición actual hasta la posición siguiente sin realizar ningún corte. Si la posición siguiente tiene una línea de comando PD, se realiza la trayectoria en línea recta desde la posición actual hasta la posición siguiente realizando el corte.
- 3.- Regresar al punto 1.

Enseguida se muestran dos dibujos realizados por medio de este procedimiento (Figura 5.9):

Dibujos trazados con Corel Draw.



Dibujos trazados con la mesa trazadora.



Figura 5.9 Comparación de dibujos con formato HPGL trazados en Corel Draw contra los trazados con la mesa trazadora.

Las imágenes trazadas con la mesa trazadora, muestran las trayectorias del corte que sigue el haz de luz láser.

# CAPITULO 6

## CONCLUSIONES Y RECOMENDACIONES.

### 6.1 Conclusiones

La necesidad de acoplar un sistema CNC en la mesa trazadora para la implementación de un sistema láser para corte de materiales suaves y las eventualidades de implementación bajo un determinado criterio, fue lo que motivó a la realización de este trabajo.

La implementación del sistema CNC en la mesa trazadora implicó el diseño de un controlador para los motores de paso en el cual se aplican conocimientos de diseño de lógica secuencial. Además se utilizan dispositivos de lógica de arreglos programable, en este caso el GAL, esto es para minimizar el tamaño del controlador.

El algoritmo para realizar las trayectorias se basa principalmente en el trazo de líneas rectas. Es decir, se descomponen todas las trayectorias en puntos muy cercanos entre sí, de acuerdo con la resolución de la mesa, y después se realiza la trayectoria mediante interpolaciones lineales. Esto no afecta a la calidad del trazo ya que cada punto tomado como muestra para la generación de la trayectoria está dentro de la escala de la misma mesa de trazos.

El control de las trayectorias es llevado mediante una computadora la cual realiza los cálculos y envía la información a los controladores mediante el puerto paralelo.

## 6.2 Recomendaciones

Como se mencionó en el apartado de conclusiones, el control de las trayectorias se llevar a cabo mediante una computadora, esto provoca que la computadora utilice gran parte de su capacidad de procesamiento en los cálculos necesarios para generar las trayectorias. Para evitar esto se pueden implementar los algoritmos para generar las trayectorias en un microcontrolador el cual se comunique por el puerto serie con una computadora. Aquí la computadora sería solamente una terminal y los cálculos serían realizados por el microcontrolador. Esto permitiría dejar recursos libres en la computadora para poder correr otras aplicaciones. El tiempo de la generación de la trayectoria sería más independiente de los recursos disponibles en la computadora.

En el controlador se deja la opción de un tercer motor a pasos para la implementación de un eje z para mover hacia arriba o abajo el láser. Ésto es para cuando se implemente el eje z y se agregue el láser al sistema.

# REFERENCIAS.

- [1] <http://www.metalunivers.com/Arees/corte/tutorial/laser.htm>
- [2] <http://www.columbia.edu/cu/mechanical/mrl/ntm/pgIndex.html>
- [3] Variable Feedrate CNC Interpolation for Planar Implicit Curves  
H.- Y. Xu, H.- Y. Tam, Z. Zhou and P. W. Tse  
Advanced Manufacturing Technology  
(2001) 18:794-800

# BIBLIOGRAFÍA.

Timothy J. Maloney.

Electrónica Industrial Moderna, 3ª. Edición.

Prentice Hall Hispanoamericana S.A. 1997.

E. E. Coherent.

LASERS. Operación, equipo, uso y diseño, 1ª. Edición.

Editorial LIMUSA, S. A. de C. V. 1985.

M. C.: Juan Ángel Garza Garza.

Manual de laboratorio de Electrónica Lógica I, 1ª. Edición.

Universidad Autónoma de Nuevo León.

Facultad de ingeniería Mecánica y Eléctrica. 2001.

Raymond A. Serway

Física (Incluye física moderna), 3ª. Edición

McGraw-Hill/Interamericanade México, S. A. de C. V. 1993.

H. M. Deitel / P. J. Deitel.

Como programar en C/C++, 2ª Edición.

Prentice Hall Hispanoamericana, S. A. 1995.

John F. Wakerly

Diseño Digital. Principios y Practicas, 1ª Edición.

Prentice Hall Hispanoamericana, S. A. 1992.

# LISTADO DE FIGURAS

Número de la Figura	Nombre	Número de Página
2.1	Motor a pasos.	9
2.2	Características T/ $\theta$ .	11
2.3	Características T/I.	12
2.4	Características dinámicas	13
2.5	Emisión estimulada	15
2.6	Elementos básicos del láser de CO <sub>2</sub>	16
2.7	Niveles de energía del sistema de CO <sub>2</sub>	17
2.8	Modelado representativo del <i>proceso de corte</i> .	17
2.9	Imagen de un archivo HPGL.	22
2.10	Brazo manipulador siguiendo las trayectorias indicadas por el archivo HPGL.	23
3.1	Sistema reductor.	26
3.2	Dimensiones y características de los engranes.	26
4.1	Diagrama general de la mesa de trazo.	31
4.2	Fotografía de la mesa de trazo.	32
4.3	Representación de las posiciones que puede tomar el haz de luz láser.	33
4.4	Desplazamientos posibles.	33
4.5	Secuencias de los motores de paso.	34
4.6	Diagrama de transición del controlador del motor a pasos.	35
4.7	Diagrama a Bloques del controlador del motor a pasos.	36
4.8	Tabla de estados del controlador del motor a pasos.	36
4.9	Asignación de valores a los estados del controlador del motor a pasos.	36

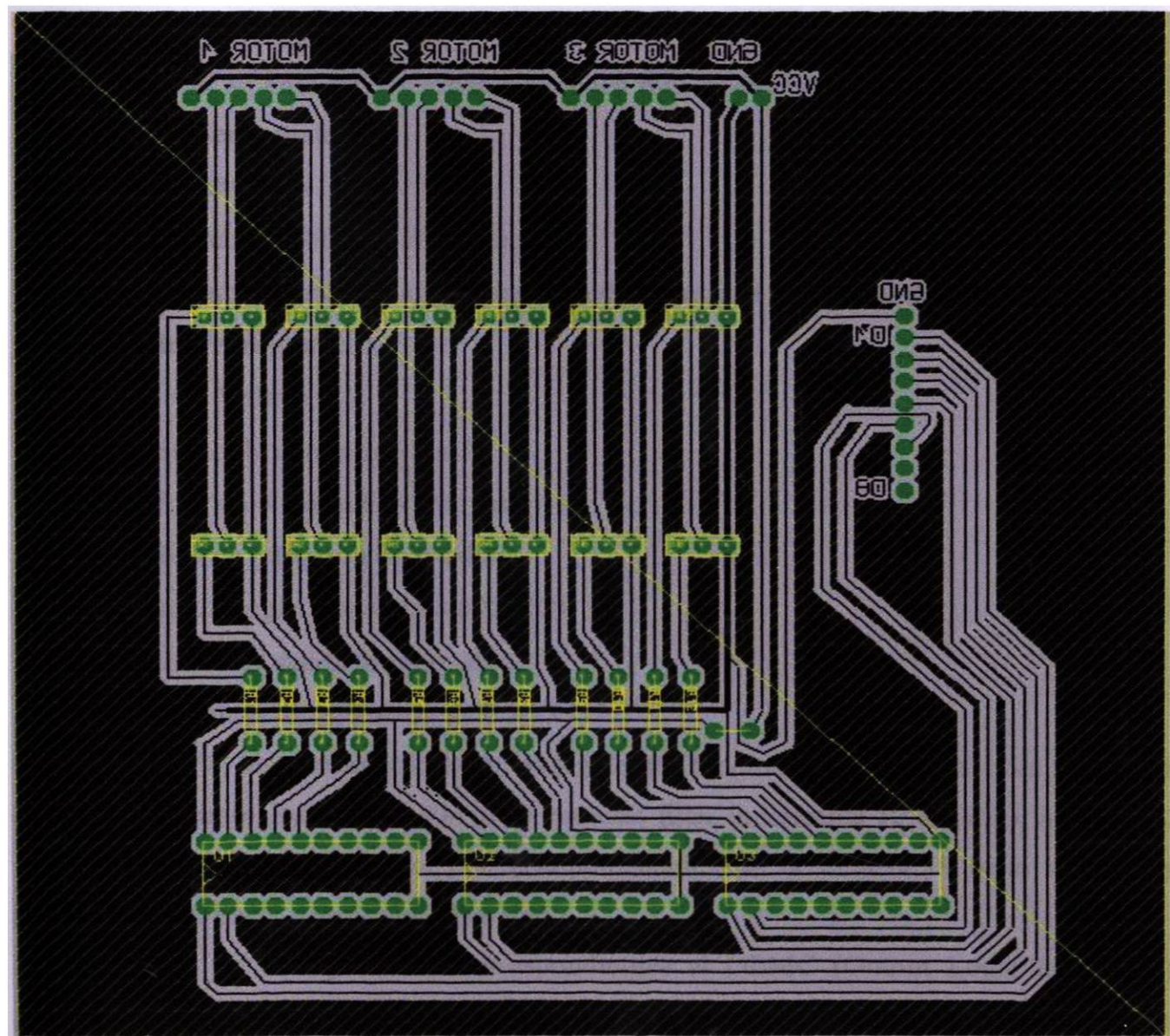
4.10	Circuito del controlador de un motor a pasos.	37
4.11	Diagrama a bloques de un controlador para 3 motores de paso.	38
4.12	Puerto paralelo (Interfase DB25).	38
4.13	Tabla de Datos utilizados en el puerto paralelo.	39
5.1	Localización del punto de referencia del sistema de coordenadas cartesianas absolutas manejadas por la mesa trazadora.	41
5.2	Sistema de coordenadas cartesianas absolutas de la mesa trazadora.	42
5.3	Sistema de coordenadas cartesianas relativas de la mesa trazadora.	42
5.4	Vectores de desplazamiento para la trayectoria en línea recta.	43
5.5	Diagrama de estados de la combinación de pulsos posibles recibidos por el controlador.	44
5.6	Componentes del círculo.	47
5.7	Diagrama de flujo para el trazo de un círculo.	48
5.8	Archivo HPGL y trayectoria seguida por la herramienta.	49
5.9	Comparación de dibujos con formato HPGL trazados en Corel Draw contra los trazados con la mesa trazadora.	50



# ANEXOS

## ANEXO A.

PC Board del controlador de los motores a pasos.



# ANEXO B.

## El GAL16V8

El GAL (Generic Array Logic), en español Arreglo Lógico Genérico, son un tipo de circuito integrado, de marca registrada por Lattice Semiconductor, que ha sido diseñado con el propósito de sustituir a la mayoría de las PAL, manteniendo la compatibilidad de sus terminales.

El GAL básicamente está formado por una matriz AND reprogramable y una matriz OR fija con configuración programable de salidas y/o entradas.

A continuación se enlistan las especificaciones más relevantes del circuito GAL16V8 de marca Lattice Semiconductor.

- $F_{max}=250$  Mhz.
- 3.5 ns máximo tiempo de propagación.
- 2.5 ns máximo tiempo de propagación de la entrada de reloj al dato de salida.
- Celdas reprogramables.
- $V_{cc} = 5$  Volts  $\pm 5\%$ .
- Consumo de corriente 90 mA.
- Rapidez en el borrado  $< 100$  ms.
- 20 años de retención de los datos.
- 8 Output Logic Macrocells (OLMC).
- Polaridad de salida programable.
- Temperatura de operación de 0 a 75 ° C.

En el GAL16V8, las terminales tienen las siguientes funciones: La terminal 1 es la entrada de CLK, las terminales 2 a la 9 son ocho entradas fijas, la terminal 10 es GND, la terminal 11 como una entrada de control O/E (Output/Enable), de la 12 a la 19 ocho terminales programables OLMC y terminal 20 es Vcc.

Las terminales de la 12 a la 19 correspondientes al OLMC (Output Logic Macrocell) pueden programarse para trabajar como entradas y/o salidas estas pueden ser

combinacionales o registradas (Flip Flops), lo cual le da la versatilidad de ser programado de diferentes formas y para diferentes requerimientos.

Programa para controlar 3 motores de pasos:

```
MODULE steep
TITLE 'secuenciador bidireccional para un motor de pasos'
c,x=.c.,.x.;
"Entradas
    Clk, UD pin 1,2;
"Salidas Combinacionales
    A,B,C,D pin 19..16 istype 'com';
"Salidas registradas
    Q1,Q0 pin 15,14 istype 'reg';
"Sincronización de los FF,s a un mismo pulso de reloj
DECLARATIONS
    sreg=[Q0,Q1];
EQUATIONS
    sreg.clk=Clk;
"Asignación de valores a los estados.
DECLARATIONS
E0=[0,0];
E1=[0,1];
E2=[1,0];
E3=[1,1];
state_diagram sreg;
STATE E0:
A=1; B=0; C=1; D=0;
if UD then E1 else E3;
STATE E1:
A=1; B=0; C=0; D=1;
```

if UD then E2 else E0;

STATE E2:

A=0; B=1; C=0; D=1;

if UD then E3 else E1;

STATE E3:

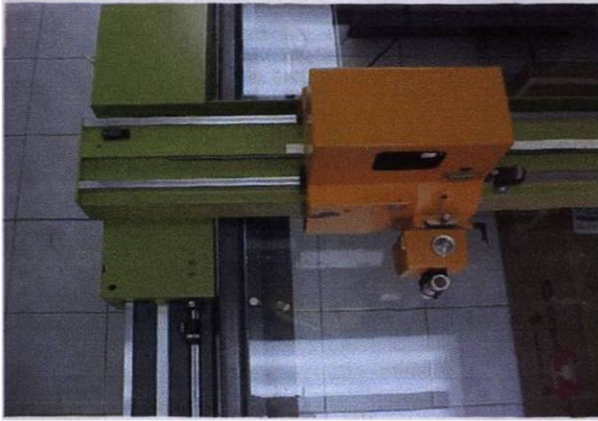
A=0; B=1; C=1; D=0;

if UD then E0 else E2;

END

# ANEXO C

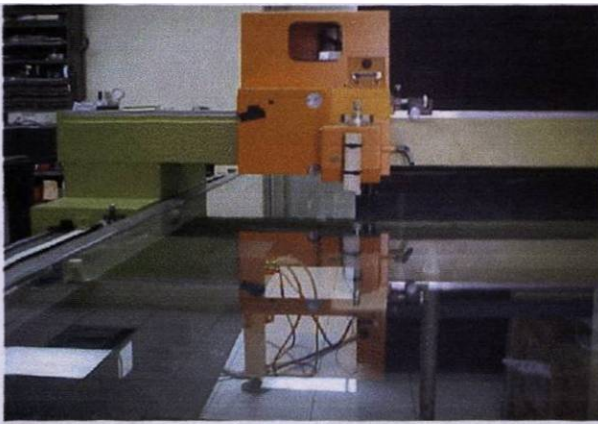
## Imágenes de la implementación.



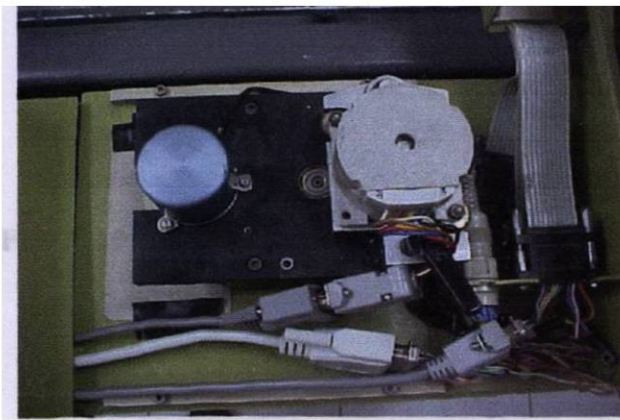
**Fig. A.1. Vista superior.**



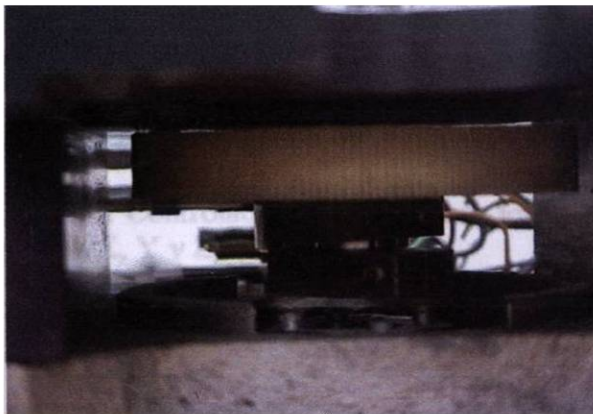
**Fig. A.2. Vista Lateral.**



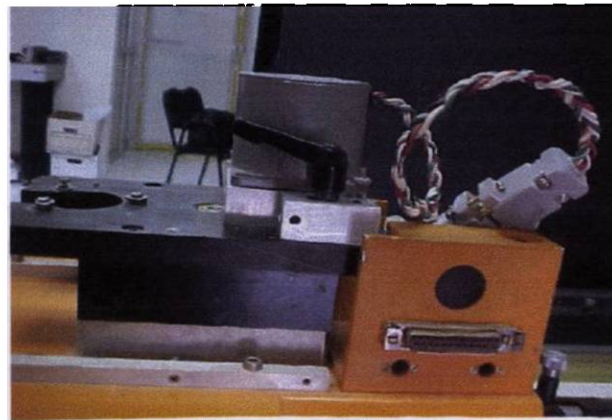
**Fig. A.3. Vista Frontal.**



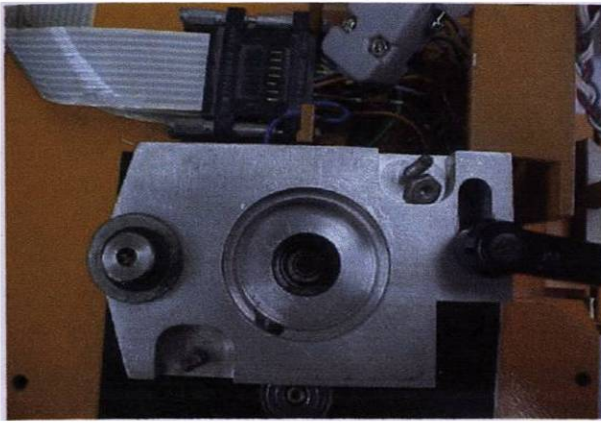
**Fig. A.4. Vista del motor "Y".**



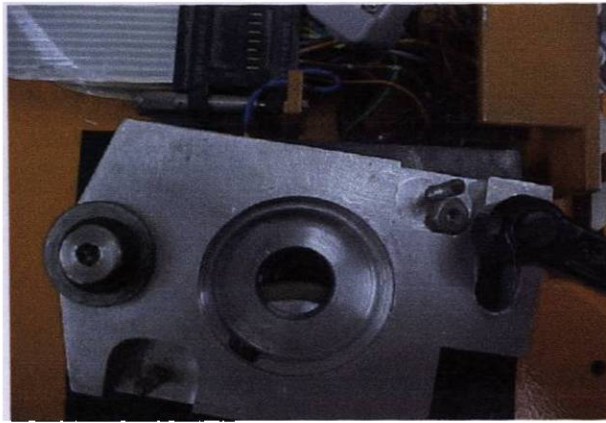
**Fig. A.5. Vista de la transmisión.**



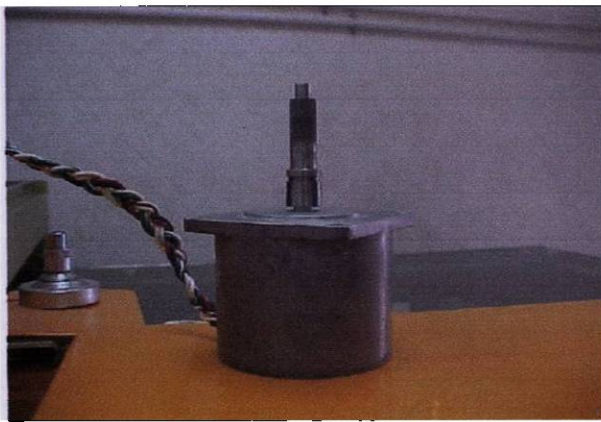
**Fig. A.6. Vista del motor "X".**



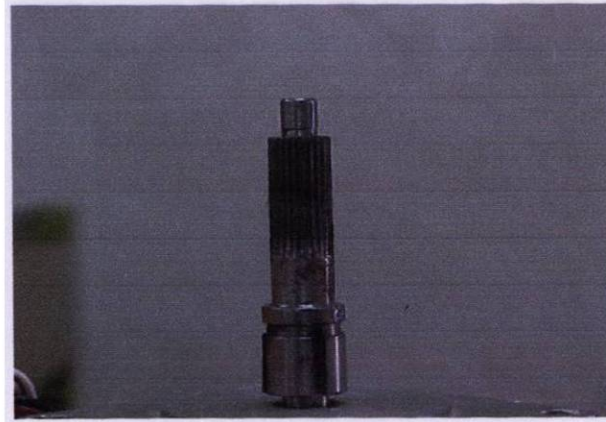
**Fig. A.7. Base del motor (Vista abierta).**



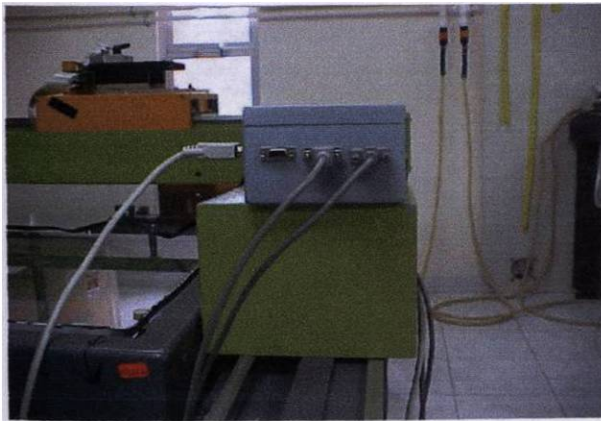
**Fig. A.8. Base del motor (Vista Cerrada).**



**Fig. A.9. Motor a pasos.**



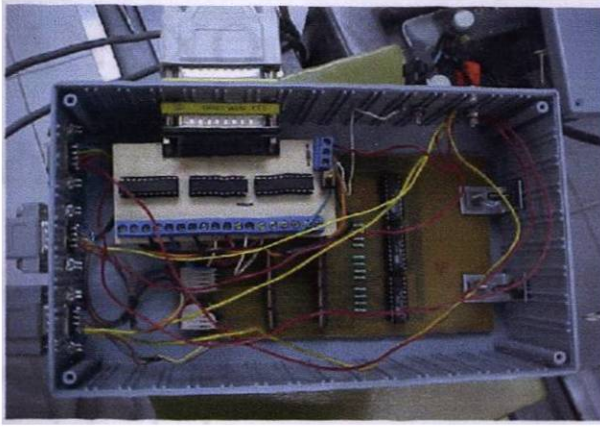
**Fig. A.10. Engrane del motor.**



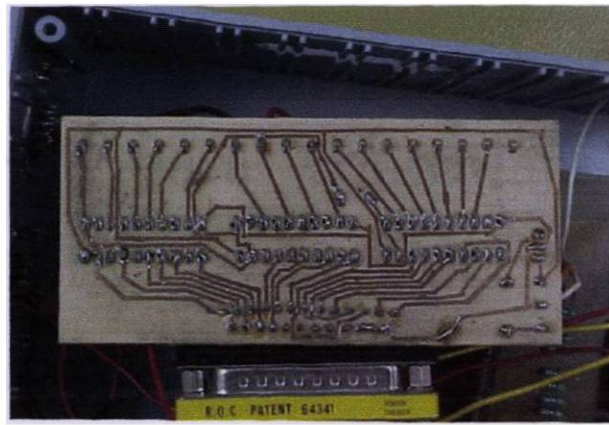
**Fig. A.11. Controlador (Conectores de los motores X, Y y Z).**



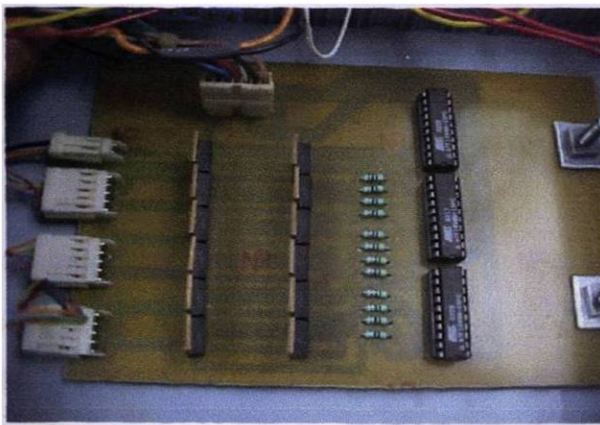
**Fig. A.12. Controlador (Alimentación y puerto DB25).**



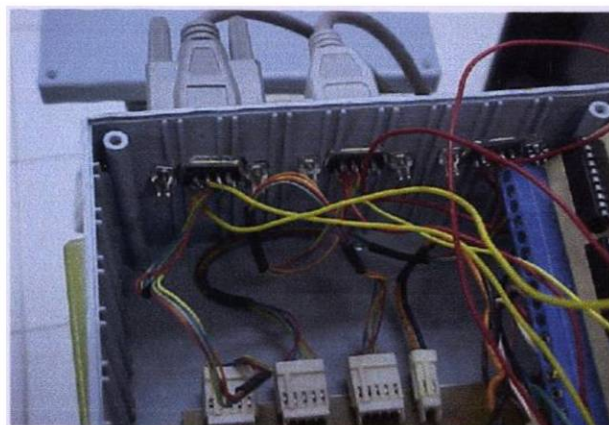
**Fig. A.13. Controlador (Vista interna).**



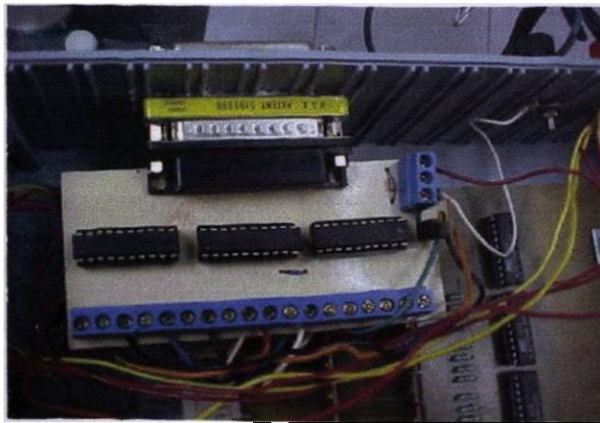
**Fig. A.14. Interfase DB25 (Vista inferior).**



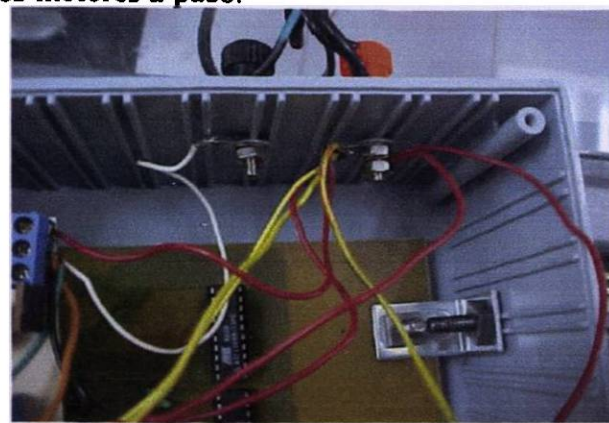
**Fig. A.15. Controlador para 3 motores a paso.**



**Fig. A.16. Interfase DB9 para las conexiones de los motores a paso.**



**Fig. A.17. Interfase DB25 (Vista superior).**



**Fig. A.18. Alimentación (5 volts, 3Amperes).**

# ANEXO D

## Programa realizado C++ para controlar la mesa trazadora.

```
#include <stdio.h>
#include <dos.h>
#include <conio.h>
#include <time.h>
#include <math.h>
#include <iostream.h>

#define DATA 0x0378
#define STATUS DATA+1
#define CONTROL DATA+2
#define RESOLUCION 0.13054
//PARAMETROS DE HPGL

FILE *p=fopen("p.plt","r");

int v,r,vc=0,rc=0;
long float xpactual, ypactual, zpactual;
long float xpanterior, ypanterior, zpanterior;

#define x0 400
#define y0 -450
#define z0 500
#define angulo .72658
#define cosa 9/sqrt(145)
#define sena 8/sqrt(145)
```



```
////////////////////////////////////  
/////////FUNCIONES DE HPGL/////////  
////////////////////////////////////
```

```
void leer_coordenadas(void);  
void inicio_de_programa(void);  
void transformacion(long int,long int);
```

```
////////////////////////////////////  
///FUNCIONES DE ESTADO//  
////////////////////////////////////
```

```
char sig;  
int my0mx0(void);  
int my0mx1(void);  
int my1mx0(void);  
int my1mx1(void);  
void inicio(void);  
void fin(void);
```

```
double xreal,yreal, dx, dy;  
long xentera,yentera,xcomparacion,ycomparacion,caso,xmaxima,ymaxima, milisegundos;
```

```
/*  
*****FUNCIONES DE ESTADOS DEL CONTROLADOR*****  
*/
```

```
int my0mx0()  
{  
delay(milisegundos);  
printf("Nodo 0 ");
```

```

int puerto;
puerto=inportb(DATA);
printf(" antes %x ",puerto);
puerto=puerto&0xFA;
outport(DATA,puerto);
printf(" valor %x ",puerto);

switch(caso)
{
case 0: return 1;

case 1:

    xreal++;
    yreal=(dy/dx)*xreal;
    yentera=yreal;
    xentera=xreal;
    printf("%ld ",xentera);printf("%ld\n",yentera);

    if(yentera>=ycomparacion)
    {
        ycomparacion++;
        if(xentera>=xmaxima&&yentera>=ymaxima)
        {
            caso=0;
            sig=0x00;
            return 1;
        }
        sig=0x11;
    }
else

```

```

{
    if(xentera>=xmaxima&&yentera>=ymaxima)
    {
        caso=0;
        sig=0x00;
        return 1;
    }
    sig=0x01;
}
break;

```

case 2:

```

yreal++;
xreal=(dx/dy)*yreal;
xentera=xreal;
yentera=yreal;
printf("%ld ",xentera);printf("%ld\n",yentera);

```

```

if(xentera>=xcomparacion)
{
    xcomparacion++;
    if(xentera>=xmaxima&&yentera>=ymaxima)
    {
        caso=0;
        sig=0x00;
        return 1;
    }
    sig=0x11;
    //my1mx10;
}
else

```

```

{
    if(xentera>=xmaxima&&yentera>=ymaxima)
    {
        caso=0;
        sig=0x00;
            // my0mx00);
        return 1;
    }
    sig=0x10;
    // my1mx00);
}
break;

```

case 3:

```

xreal++;
yreal++;
xentera=xreal;
yentera=yreal;
//printf("%d ",xentera);printf("%f\n",xreal);
printf("%ld ",xentera);printf("%ld\n",yentera);
if(xentera>=xmaxima||yentera>=ymaxima)
    {
        caso=0;
        sig=0x00;
        //my0mx00);
        return 1;
    }
sig=0x11;
//my1mx10);

```

```
break;
default: fin();break;
}
return 1;

}
```

```
int my0mx1()
{
delay(milisegundos);
printf("Nodo 1 ");
```

```
int puerto;
puerto=inportb(DATA);
printf(" antes %x ",puerto);
puerto=puerto&0xFB;
puerto=puerto|0x01;
outport(DATA,puerto);
printf(" valor %x ",puerto);
```

```
switch(caso)
{
case 0: return 1;
```

```
case 1:
    xreal++;

    yreal=(dy/dx)*xreal;
    yentera=yreal;
    xentera=xreal;
```

```
printf("%ld ",xentera);printf("%ld\n",yentera);
```

```
if(yentera>=ycomparacion)
```

```
{
```

```
    ycomparacion++;
```

```
    if(xentera>=xmaxima&&yentera>=ymaxima)
```

```
    {
```

```
        caso=0;
```

```
        sig=0x00;
```

```
        //my0mx00);
```

```
        return 1;
```

```
    }
```

```
        sig=0x10;
```

```
        //my1mx00);
```

```
}
```

```
else
```

```
{
```

```
    if(xentera>=xmaxima&&yentera>=ymaxima)
```

```
    {
```

```
        caso=0;
```

```
        sig=0x00;
```

```
        //my0mx00);
```

```
        return 1;
```

```
    }
```

```
        sig=0x00;
```

```
        //my0mx00);
```

```
}
```

```
//fin();
```

```
break;
```

case 2:

```

yreal++;
xreal=(dx/dy)*yreal;
xentera=xreal;
yentera=yreal;
printf("%ld ",xentera);printf("%ld\n",yentera);

if(xentera>=xcomparacion)
{
    xcomparacion++;
    if(xentera>=xmaxima&&yentera>=ymaxima)
    {
        caso=0;
        sig=0x00;
        // my0mx0();
        return 1;
    }
    sig=0x10;
    //my1mx0();
}
else
{
    if(xentera>=xmaxima&&yentera>=ymaxima)
    {
        caso=0;
        sig=0x00;
        // my0mx0();
        return 1;
    }
    sig=0x10;
    //my1mx0();
}

```

```
    }  
    break;
```

```
case 3:
```

```
    xreal++;  
    yreal++;  
    xentera=xreal;  
    yentera=yreal;  
    printf("%ld ",xentera);printf("%ld\n",yentera);  
    if(xentera>=xmaxima||yentera>=ymaxima)  
        {  
            caso=0;  
            sig=0x00;  
            //my0mx0();  
            return 1;  
        }  
    sig=0x11;  
    //my1mx1();
```

```
break,  
default: fin();break;  
}
```

```
return 1;  
}
```

```
int my1mx0()  
{  
delay(miliseundos);
```



```
printf("Nodo 2 ");
```

```
int puerto;
```

```
puerto=inportb(DATA);
```

```
printf(" antes %x ",puerto);
```

```
puerto=puerto&0xFE;
```

```
puerto=puerto|0x04;
```

```
outport(DATA,puerto);
```

```
printf(" valor %x ",puerto);
```

```
switch(caso)
```

```
{
```

```
case 0: return 1; //break;
```

```
case 1:
```

```
    xreal++;
```

```
    yreal=(dy/dx)*xreal;
```

```
    yentera=yreal;
```

```
    xentera=xreal;
```

```
    printf("%ld ",xentera);printf("%ld\n",yentera);
```

```
    if(yentera>=ycomparacion)
```

```
    {
```

```
        ycomparacion++;
```

```
        if(xentera>=xmaxima&&yentera>=ymaxima)
```

```
        {
```

```
            caso=0;
```

```
            sig=0x00;
```

```
            //my0mx00);
```

```

        return 1;
    }
    sig=0x01;
    //my0mx10);
}
else
{
    if(xentera>=xmaxima&&yentera>=ymaxima)
    {
        caso=0;
        sig=0x00;
        //my0mx00);
        return 1;
    }
    sig=0x01;
    //my0mx10);
}
//fin();
break;

```

case 2:

```

yreal++;
xreal=(dx/dy)*yreal;
xentera=xreal;
yentera=yreal;
printf("%ld ",xentera);printf("%ld\n",yentera);

if(xentera>=xcomparacion)
{
    xcomparacion++;
    if(xentera>=xmaxima&&yentera>=ymaxima)

```

```

    {
    caso=0;
    sig=0x00;
    //my0mx0();
    return 1;
    }
    sig=0x01;
    //my0mx1();
}
else
{
    if(xentera>=xmaxima&&yentera>=ymaxima)
    {
    caso=0;
    sig=0x00;
    //my0mx0();
    return 1;
    }
    sig=0x00;
    //my0mx0();
}
break;

```

case 3:

```

xreal++;
yreal++;
xentera=xreal;
yentera=yreal;
printf("%ld ",xentera);printf("%ld\n",yentera);
if(xentera>=xmaxima||yentera>=ymaxima)

```

```

        {
        caso=0;
        sig=0x00;
        //my0mx00);
        return 1;
        }
    sig=0x11;
    //my1mx10);
    break;
default: fin();break;
}

```

```

return 1;
}

```

```

int my1mx10)
{
delay(miliseundos);
printf("Nodo 3 ");

```

```

int puerto;
puerto=inportb(DATA);
printf(" antes %x ",puerto);
puerto=puerto|0x05;
outport(DATA,puerto);
printf(" valor %x ",puerto);

```

```

switch(caso)
{
case 0: return 1; //break;

```

case 1:

```
xreal++;  
yreal=(dy/dx)*xreal;  
yentera=yreal;  
xentera=xreal;  
printf("%ld ",xentera);printf("%ld\n",yentera);
```

```
if(yentera>=ycomparacion)  
{  
    ycomparacion++;  
    if(xentera>=xmaxima&&yentera>=ymaxima)  
    {  
        caso=0;  
        sig=0x00;  
        //my0mx0();  
        return 1;  
    }  
    sig=0x00;  
    //my0mx0();  
}  
else  
{  
    if(xentera>=xmaxima&&yentera>=ymaxima)  
    {  
        caso=0;  
        sig=0x00;  
        //my0mx0();  
        return 1;  
    }  
    sig=0x00;
```

```
        //my0mx00);
    }
//fin();
break;
```

case 2:

```
yreal++;
xreal=(dx/dy)*yreal;
xentera=xreal;
yentera=yreal;
printf("%ld ",xentera);printf("%ld\n",yentera);
```

```
if(xentera>=xcomparacion)
```

```
{
    xcomparacion++;
    if(xentera>=xmaxima&&yentera>=ymaxima)
    {
        caso=0;
        sig=0x00;
        //my0mx00);
        return 1;
    }
    sig=0x00;
    //my0mx00);
```

```
}
```

```
else
```

```
{
    if(xentera>=xmaxima&&yentera>=ymaxima)
    {
        caso=0;
        sig=0x00;
```

```
        //my0mx00);
        return 1;
    }
    sig=0x00;
    //my0mx00);
}
break;
```

case 3:

```
xreal++;
yreal++;
xentera=xreal;
yentera=yreal;
printf("%ld ",xentera);printf("%ld\n",yentera);
if(xentera>xmaxima||yentera>ymaxima)
    {
        caso=0;
        sig=0x00;
        //my0mx00);
        return 1;
    }
sig=0x00;
//my0mx00);
```

```
break;
default: fin();break;
```

```
}
```

```
return 1;
}
```

```
/*  
*****MOTORES INDEPENDIENTES*****  
*****/
```

```
int motor1(int sentido1)
{
    if(sentido1==0)
    {
        outportb(DATA,0x03);
        delay(miliseundos);
        outportb(DATA,0x02);
        delay(miliseundos);
        return 1;
    }

    if(sentido1==1)
    {
        outportb(DATA,0x01);
        delay(miliseundos);
        outportb(DATA,0x00);
        delay(miliseundos);
        return 1;
    }
    return 0;
}
```

```
int motor2(int sentido2)
```



```
{  
  
    if(sentido2==0)  
    {  
        outportb(DATA,0x04);  
        delay(milisegundos);  
        outportb(DATA,0x00);  
        delay(milisegundos);  
        return 1;  
    }  
    if(sentido2==1)  
    {  
        outportb(DATA,0xC);  
        delay(milisegundos);  
        outportb(DATA,0x08);  
        delay(milisegundos);  
        return 1;  
    }  
    return 0;  
}
```

```
int motor3(int sentido3)
```

```
{  
  
    if(sentido3==0)  
    {  
        outportb(DATA,0x10);  
        delay(milisegundos);  
        outportb(DATA,0x00);  
        delay(milisegundos);  
        return 1;  
    }  
}
```

```

    }
    if(sentido3==1)
    {
        outportb(DATA,0x30);
        delay(miliseundos);
        outportb(DATA,0x20);
        delay(miliseundos);
        return 1;
    }
    return 0;
}

```

```

/*****
*****FUNCION INICIO*****
*****/

```

```

void inicio(void)
{

if(dx>dy)
    caso=1;
else
if(dx<dy)        //SELECCIONA EL CASO DE LA PENDIENTE
    caso=2;        //1 si dx>dy. 2 si dx<dy. 3 si dx=dy.
else
if(dx==dy)
    caso=3;

/*
if(dx>5000)

```

```

{
switch(caso)
{
caso 1:
dxmax=dx;
dymax=dy;
c=0;
do
{
dx=5000;
dy=dymax*5000/dxmax;

xreal=0;
xcomparacion=1;

yreal=0;
ycomparacion=1;

xmaxima=dx;
ymaxima=dy;

printf("%d\n", caso);
my0mx0(); //inicia la secuencia de estados

c++;
}else((c-1)*5000+<)

}

} */

```

```

sig=0x00;
printf("%d\n", caso);
while(caso)
{ switch(sig)
{
case 0x00: my0mx0();
        break;
case 0x01: my0mx1();
        break;
case 0x10: my1mx0();
        break;
case 0x11: my1mx1();
        break;
default: break;//chekar errores
}
} //while          //inicia la secuencia de estados
} //inicio

/*****
*****FUNCION FIN*****
*****/

void fin(void)
{
printf("%lf\n%d", xreal, caso);
//getche();
}

```

```
/******  
*****FUNCION SENTIDOS()*****  
*****/
```

```
int sentidoX(long x1, long x2)
```

```
{  
int puerto, sentidox;  
if(x2-x1<0)  
    {  
        sentidox=0;  
        puerto=inportb(DATA);  
        puerto=puerto|0x02;  
        outportb(DATA,puerto);  
    }  
else  
    { sentidox=1;  
        puerto=inportb(DATA);  
        puerto=puerto&0xFD;  
        outportb(DATA,puerto);  
    }  
if(x2-x1==0)  
sentidox=2;  
  
return sentidox;  
}
```

```
int sentidoY(long y1, long y2)
```

```
{  
int puerto, sentidoy;  
if(y2-y1<0)
```

```

    {
        sentidoy=0;
        puerto=inportb(DATA);
        puerto=puerto&0xF7;
        outportb(DATA,puerto);
    }
    else
    { sentidoy=1;
        puerto=inportb(DATA);
        puerto=puerto|0x08;
        outportb(DATA,puerto);
    }
    if(y2-y1==0)
    sentidoy=2;
    return sentidoy;
}

```

```

/*****
*****CALCULARPARAMETROSRECTA()*****
*****/

```

```

calcularparametrosrecta(long x1,long x2,long y1,long y2)

```

```

{
    long int valor, sentidox, sentidoy, pasos;
    clrscr();
    long absx,absy;
    absx = (x2>x1) ? x2-x1 : x1-x2;
    absy = (y2>y1) ? y2-y1 : y1-y2;

    dx=2*absx;

```

```

dy=2*absy;
xreal=0;
xcomparacion=1;
yreal=0;
ycomparacion=1;
xmaxima=dx;
ymaxima=dy;

//milisegundos=1;

/*SENTIDOS*/

sentidox=sentidoX(x1,x2);
sentidoy=sentidoY(y1,y2);
printf("\nSentidox=%d",sentidox);
printf("\nSentidoy=%d",sentidoy);

if(sentidox==2&&sentidoy!=2)
{
clock_t start, end;
start = clock();

for(pasos=0;pasos<=abs(y2-y1);pasos++)
valor=motor2(sentidoy);

end = clock();
printf("El tiempo motor y fue: %f\n", (end - start) / CLK_TCK);
fin();
}

```

```

if(sentidoy==2&&sentidox!=2)
{
clock_t start, end;
start = clock();

for(pasos=0;pasos<=abs(x2-x1);pasos++)
valor=motor1(sentidox);

end = clock();
printf("El tiempo x fue: %f\n", (end - start) / CLK_TCK);
fin();
}

if(sentidoy!=2&&sentidox!=2)
{
clock_t start, end;
start = clock();

inicio();

end = clock();
printf("El tiempo fue: %f\n", (end - start) / CLK_TCK);
fin();
}
return 0;
}

```



```

/*****
*****FUNCION PEDIRCOORDENADAS()*****
*****/

int pedircoordenadas(char opcion)
{

double x1, x2, y1, y2;
printf("\nPunto inicial");
printf("\nx1 = ");
scanf("%lf",&x1);
printf("\ny1 = ");
scanf("%lf",&y1);
printf("\nPunto final");
printf("\nx2 = ");
scanf("%lf",&x2);
printf("\ny2 = ");
scanf("%lf",&y2);
printf("\nDame el tiempo entre cada pulso: ");
cin>>milisegundos;

switch(opcion)
{
case '1': calcularparametrosrecta(long(x1),long(x2),long(y1),long(y2));return 1;//break;
default: return 0;
}

}

```

```

/*****
*****FUNCION CIRCULO*****
*****/

void circulo(void)
{
long puntodeiniciox, puntodeinicioy;
double radio,theta,theta0, xcirculoanterior, ycirculoanterior,xcirculoactual, ycirculoactual;

printf("\nRadio del circulo? ");
cin>>radio;
printf("\nPunto de inicio x? ");
scanf("%ld",&puntodeiniciox);
printf("\nPunto de inicio y? ");
scanf("%ld",&puntodeinicioy);

printf("\nDame el tiempo entre pulsos: ");
cin>>milisegundos;

theta0=RESOLUCION/radio;
theta=0;
xcirculoactual=radio*cos(theta);
ycirculoactual=radio*sin(theta);

do
{
theta=theta+theta0;
xcirculoanterior=xcirculoactual;
ycirculoanterior=ycirculoactual;
xcirculoactual=radio*cos(theta);
ycirculoactual=radio*sin(theta);
}

```

```
calcularparametrosrecta(long(xcirculoanterior),long(xcirculoactual),long(ycirculoanterior),l  
ong(ycirculoactual));
```

```
printf("%lf\n",theta);
```

```
}while(theta<2*3.141592);
```

```
getche();
```

```
}
```

```
/*  
*****  
*****FUNCIONES DE HPGL*****  
*****  
*/
```

```
void leer_coordenadas(void)
```

```
{
```

```
long int x,y;
```

```
xpanterior=xpactual;
```

```
ypanterior=ypactual;
```

```
zpanterior=zpactual;
```

```
fscanf(p,"%ld",&x);
```

```
fscanf(p,"%ld",&y);
```

```
xpactual=0.5*x;
```

```
ypactual=0.5*y;
```

```
transformacion(xpactual,ypactual);
```

```
}
```

```

void transformacion(long int x,long int y)
{
long int z=0;
xpactual=x*cosa-y*sena+x0;
ypactual=x*sena+y*cosa+y0;
zpactual=z+z0;
}

```

```

int archivohpgl(void)

```

```

{
char b;
int a,x,y,c;

xpactual=0;
ypactual=0;
zpactual=0;
printf("\nDame el tiempo entre pulsos: ");
cin>>milisegundos;

```

```

for(a=0;;a++)

```

```

{
fscanf(p,"%c",&b);
if(b=='P')
    {fscanf(p,"%c",&b);
    if(b=='D')
        {
            leer_coordenadas();

```

```

calcularparametrosrecta(long(xpanterior),long(xpactual),long(ypanterior),long(ypactual));
        }
    if(b=='U')

```

```

        {
            leer_coordenadas();

            calcularparametrosrecta(long(xpanterior),long(xpactual),long(ypanterior),long(ypactual));
        }
    }
if(b=='S')
    {fscanf(p,"%c",&b);
        if(b=='P')
            {fscanf(p,"%d",&c);
                switch (c)
                {
                    case 0:
                        return 0;
                }
            }
        if(b=='0')
            {return 0;}
        }
    }
}
}
}

```

```

/*****
*****FUNCION MENU()*****
*****/

```

```

void menu(void)
{

```

```

int error;
char opcion;

do{
clrscr();
printf("ORCA SOFTWARE. MESA TRAZADORA Versi3n BETA\n\n\n\n");
printf("1.-Recta\n");
printf("2.-Circulo\n");
printf("3.-Archivo\n");
printf("4.-Salir\n\n\n\n\n");
printf("Escoge una opci3n: ");
opcion=getche();
clrscr();
switch(opcion)
{
case '1':
printf("RECTA");pedircoordenadas(opcion);//recta();
break;
case '2': printf("CIRCULO");circulo();
break;
case '3': printf("ARCHIVO HPGL");error=archivohpgl();
break;
case '4': printf("SALIR");
break;
default: printf("No es opci3n valida");break;
}
}while(opcion!='4');
}

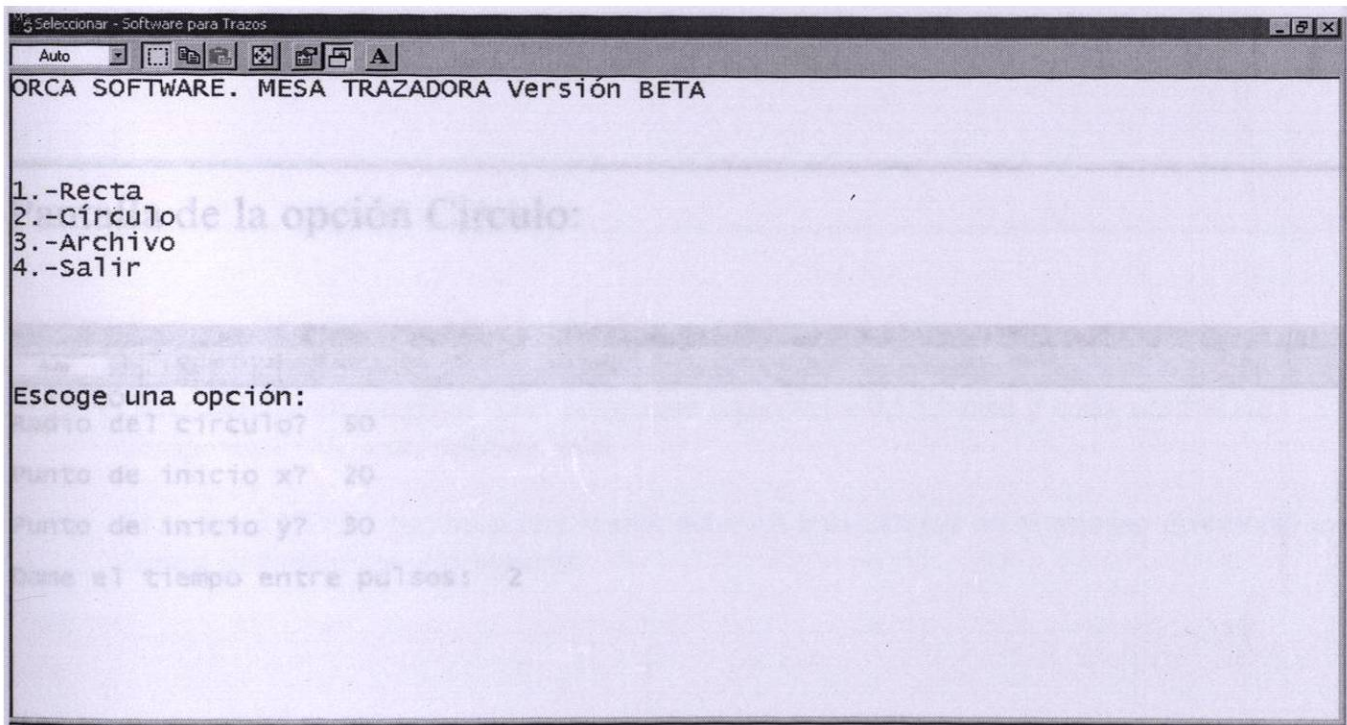
void main(void)

```

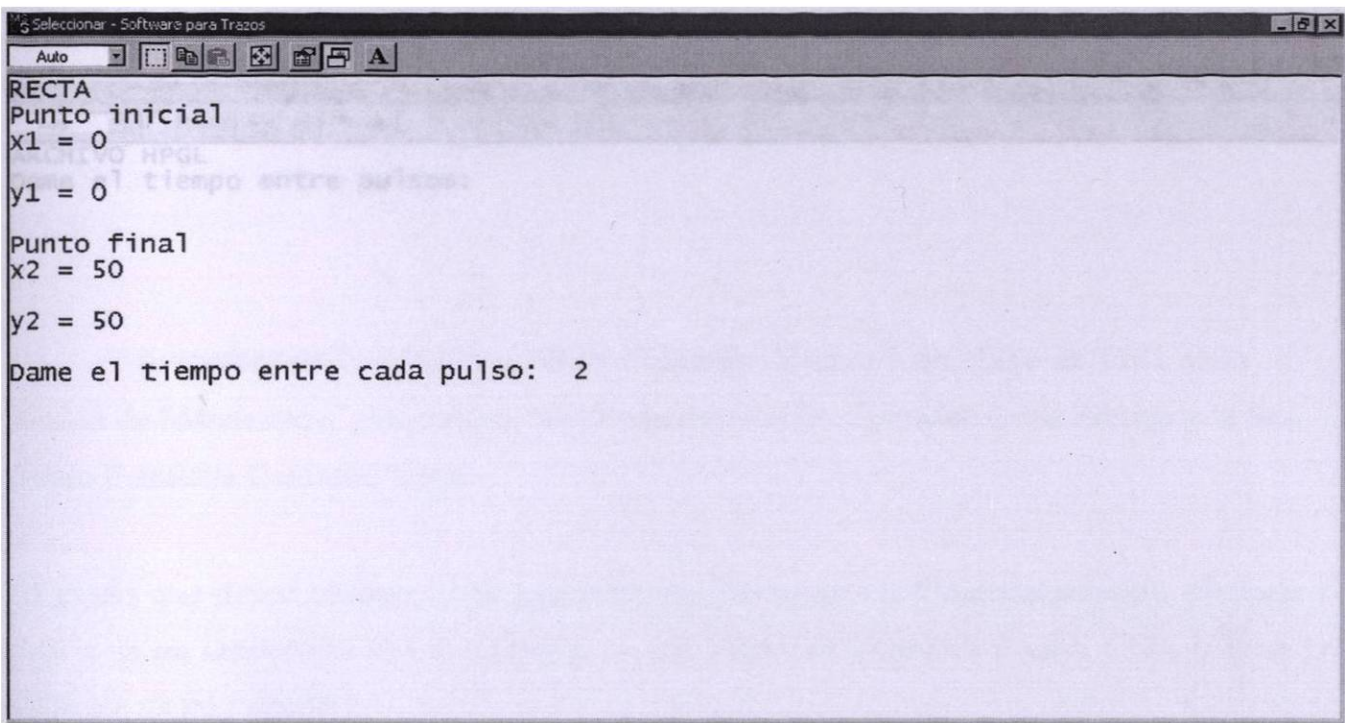
```
{  
clrscr();  
menu();  
getche();  
}
```

## Pantallas del programa:

### Pantalla del Menú:

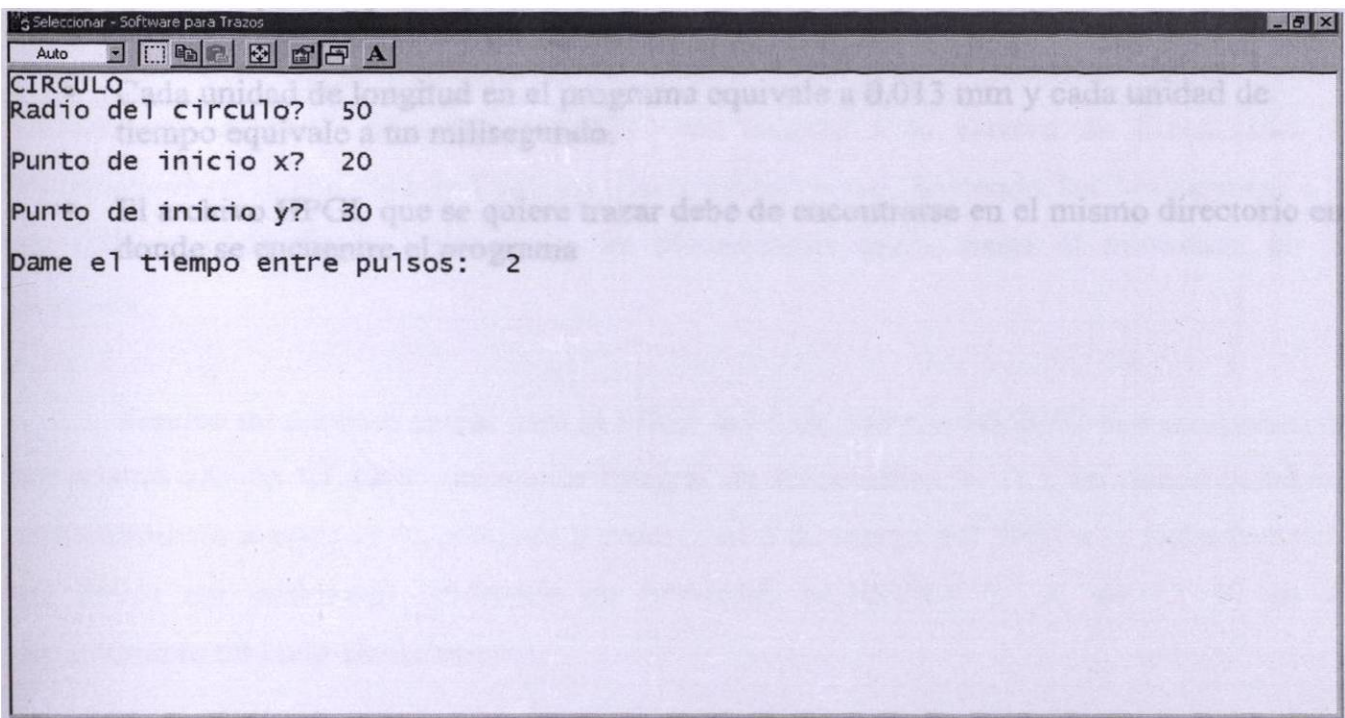


## Pantalla de la opción Recta:



```
Seleccíonar - Software para Trazos
Auto
RECTA
Punto inicial
x1 = 0
y1 = 0
Punto final
x2 = 50
y2 = 50
Dame el tiempo entre cada pulso: 2
```

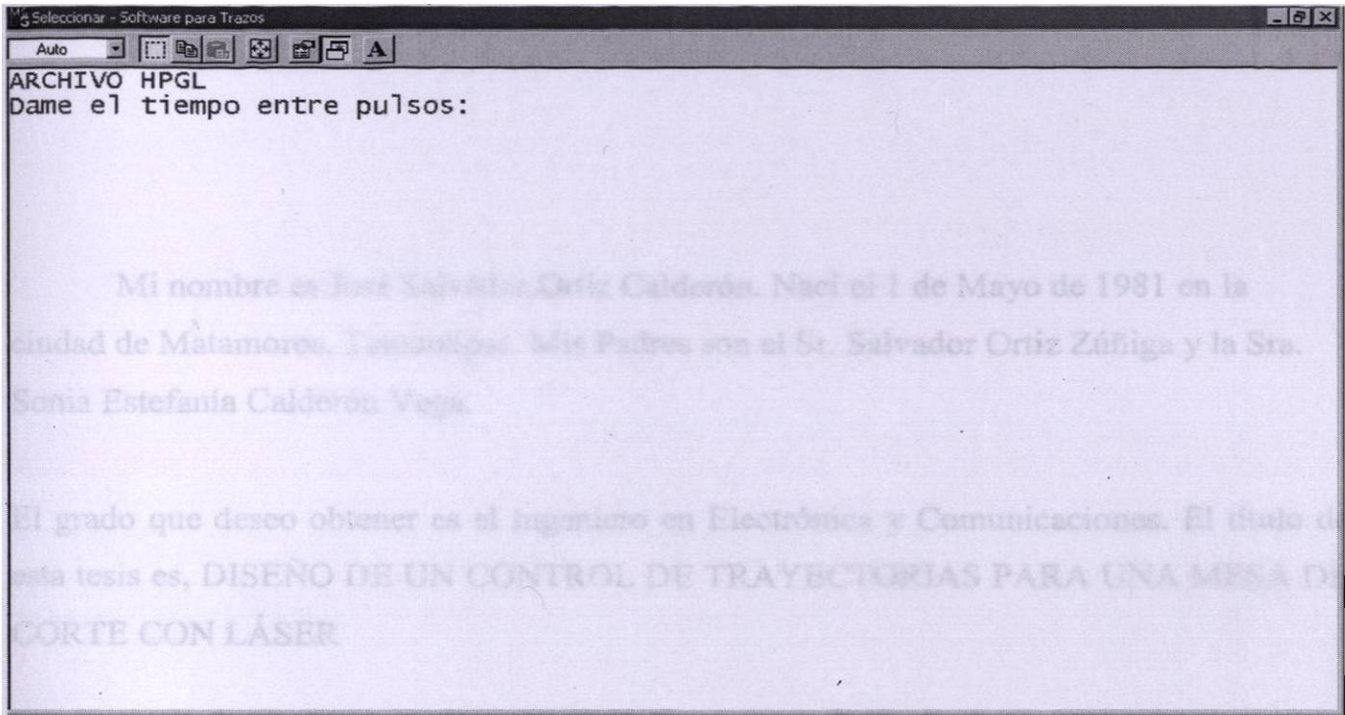
## Pantalla de la opción Círculo:



```
Seleccíonar - Software para Trazos
Auto
CIRCULO
Radio del círculo? 50
Punto de inicio x? 20
Punto de inicio y? 30
Dame el tiempo entre pulsos: 2
```



## Pantalla de la opción Archivo:



### Comentarios sobre el programa.

- Cada unidad de longitud en el programa equivale a 0.013 mm y cada unidad de tiempo equivale a un milisegundo.
- El archivo HPGL que se quiere trazar debe de encontrarse en el mismo directorio en donde se encuentre el programa

# AUTOBIOGRAFÍA

Mi nombre es José Salvador Ortiz Calderón. Nací el 1 de Mayo de 1981 en la ciudad de Matamoros, Tamaulipas. Mis Padres son el Sr. Salvador Ortiz Zúñiga y la Sra. Sonia Estefanía Calderón Vega.

El grado que deseo obtener es el Ingeniero en Electrónica y Comunicaciones. El título de esta tesis es, DISEÑO DE UN CONTROL DE TRAYECTORIAS PARA UNA MESA DE CORTE CON LÁSER

Hice mis estudios de preparatoria en la Preparatoria # 7 de la Universidad Autónoma de Nuevo León y realicé mis estudios en Ingeniería en Electrónica y Comunicaciones en la Facultad de Ingeniería Mecánica y Eléctrica de la misma Universidad. A partir de enero del 2001 me inscribí a la carrera de Licenciado en Matemáticas en la Facultad de Ciencias Físico Matemáticas, haciendo las dos carreras a la par (IEC y LM). En la Licenciatura en Matemáticas estoy, hasta el momento, en 6º Semestre.

Realice mi servicio social para la FIME del 4 de febrero del 2002 al 4 de agosto de ese mismo año en CIISA (Consultaría Integral de Informática S. A.), en donde brindaba mantenimiento a equipos de cómputo y redes. Del 3 de marzo del 2003 a la fecha (octubre del 2003) me encuentro laborando en Servicios ALESTRA S. A. de C. V. en el departamento de Help Desk Internet.

