



# UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

## Desarrollo de un sistema de control para tres servomotores con interfase a una PC

T E S I S

QUE PARA OBTENER EL TÍTULO DE:  
INGENIERO ELÉCTRICO  
ELECTRÓNICO

PRESENTA

**Amílcar Daniel Perea Matsumura**

DIRECTOR DE TESIS: M.I. BENJAMÍN VALERA OROZCO



MEXICO, D.F.

Noviembre de 2005

## AGRADECIMIENTOS

*Este es sin duda un espacio que no quería desaprovechar para mostrar mi más grande aprecio, respeto, admiración y mostrarle mi gratitud a:*

Mi madre, por ser el *motor* por el cual hoy he llegado hasta esta instancia. Bien sabes que te estaré sumamente agradecido, no sólo por esta etapa, sino porque a lo largo de toda mi existencia has sabido apoyarme y motivarme para todo lo que hasta ahora he logrado.

Mi padre, por ser el *generador* y parte fundamental de lo que ahora soy. Siempre te agradeceré la confianza y la incondicionalidad para que mis objetivos sean alcanzados. Nunca olvides lo importante que has sido para mí a lo largo de mi desenvolvimiento.

Mi hermano, por ser ese *engrane* irremplazable, el cual se ha sabido adaptar y embonar y al cual le agradezco no sólo por su comprensión, sino también por estar ahí conmigo en las buenas y en las malas y ser una persona en la que siempre podré confiar.

Mi director de Tesis, *fuerza* que me impulsó y me brindó lo necesario para desarrollar este trabajo y con el cual podré concluir esta fase, que es de suma importancia en mi vida personal y en mi actividad profesional.

Mis tías(os), ya que han sido esos *conductores* que me han sabido llevar y aconsejar en los momentos en que más los necesité.

Mis primas(os), en quienes he encontrado esa *conexión* que ha sido fundamental y de quienes siempre he recibido algo más que afecto y apoyo.

Mis amigas(os), elementos *transformadores* que han influido en mi vida y personalidad, siendo testigos de mi evolución, de mis errores y por supuesto de mis logros.

Mis profesores en general, cuya *transmisión* de conocimientos ha sido más que dadivosa y me han formado desde los primeros años de mi trayectoria.

Ese ser especial en mi vida en cuya figura encontré la inspiración y la paciencia necesarias en este periodo de mi desarrollo. Espero que esa *inducción* sea mutua y nos mantenga siempre unidos, aunque no sea de forma física.

Todas aquellas personas que, directa o indirectamente se han sabido *acoplar* y han podido compartir mis intereses y me han demostrado su cariño leal y sincero.

**A todos y cada uno de ustedes, muchas gracias por ser un elemento insustituible para que este gran *circuito* haya funcionado correctamente y así continúe...**

**AMÍLCAR DANIEL PEREA MATSUMURA**

# ÍNDICE TEMÁTICO

<b>Introducción</b>	<b>1</b>
<b>Objetivos</b>	<b>1</b>
<b>Definición del problema</b>	<b>2</b>
Antecedentes	3
Descripción del problema a resolver	5
Relevancia y justificación	5
Alcance y limitaciones	6
Relación con otras áreas	6
<b>Método</b>	<b>6</b>
<b>Resultados esperados</b>	<b>7</b>
<b>Resumen de la tesis</b>	<b>7</b>
<b>1. Antecedentes</b>	<b>9</b>
<b>1.1. Máquinas de Medir por Coordenadas</b>	<b>10</b>
1.1.1. Entendiendo la MMC	13
1.1.2. Técnicas efectivas del sensor	22
1.1.3. Parametrización de elementos geométricos en una MMC	25
<b>1.2. Codificadores ópticos</b>	<b>26</b>
<b>1.3. Control de movimiento en motores CD</b>	<b>29</b>
1.3.1. Modelado de motores CD	29
1.3.2. Control Digital	32

---

1.3.3.	Control PID	33
<b>2.</b>	<b>Sistema propuesto</b>	<b>39</b>
2.1.	Esquema general	39
2.2.	Descripción hardware	41
2.2.1.	Circuitos y módulos para el control de motores CD	42
2.2.2.	Módulo de control	42
2.2.3.	Adaptador para puerto serie	44
2.2.4.	Palpador	45
2.2.5.	Prototipo electro-mecánico	46
2.3.	Descripción software para el operador	48
2.3.1.	Pantalla principal	50
2.3.2.	Menú Archivo	52
2.3.3.	Menú Ver	53
2.3.4.	Menú Ayuda	53
2.4.	Descripción software para el programador	55
2.4.1.	Clase CAboutDlg	55
2.4.2.	Clase CCambiarPosX	56
2.4.3.	Clase CCambiarPosY	56
2.4.4.	Clase CCambiarPosZ	57
2.4.5.	Clase CmainFrame	57
2.4.6.	Clase CMMCApp	58
2.4.7.	Clase CMMCDoc	58
2.4.8.	Clase CMMCView	59
<b>3.</b>	<b>Resultados y conclusiones</b>	<b>65</b>
3.1.	Resultados	65
3.1.1.	Repetibilidad en el posicionamiento	66
3.1.2.	Repetibilidad en el punto de contacto	74

<b>3.2. Conclusiones</b>	<b>86</b>
<b>3.3. Trabajo a futuro</b>	<b>88</b>
<b>Anexos</b>	<b>89</b>
<b>A.1. Diagramas mecánicos</b>	<b>89</b>
<b>A.2. Diagramas electrónicos</b>	<b>93</b>
<b>A.3. Diagramas de los conectores</b>	<b>94</b>
<b>Referencias</b>	<b>97</b>

# INTRODUCCIÓN

## Objetivos

El presente trabajo de tesis es la continuación a un proyecto que se ha realizado al cabo de casi 15 años en el Centro de Ciencias Aplicadas y Desarrollo Tecnológico (CCADET) de la UNAM, en el que se ha desarrollado una Máquina de Medición por Coordenadas (MMC).

Estos instrumentos de medición geométrica son importantes en la industria debido a la exactitud y precisión en los datos que generan de los objetos medidos. Sin embargo, esta tecnología hasta la fecha es costosa y por ende sólo es accesible para empresas grandes con el capital suficiente como para invertir en una máquina de estas características, dejando sin posibilidad de aprovechar dichas ventajas a la industria nacional.

Dadas las condiciones anteriores y debido a que en México existe un rezago en éste y en muchos campos más, se cree que es importante el desarrollo de una máquina que reúna las características mencionadas para contribuir con diversos procesos industriales acordes con la situación y necesidades nacionales.

En particular, los objetivos planteados para la presente tesis, se describen a continuación:

- Desarrollar hardware y software para el control de posición y velocidad de tres servomotores para su uso en la transmisión de movimiento a máquinas de hasta tres ejes coordenados.
- Programar y gobernar las funciones que deberán realizar los motores que integran la MMC mediante una interfase PC.

- Crear una interfase de control y adquisición de datos para que el operador tenga un mayor dominio de la máquina y un menor porcentaje de error al usar el dispositivo.

## Definición del problema

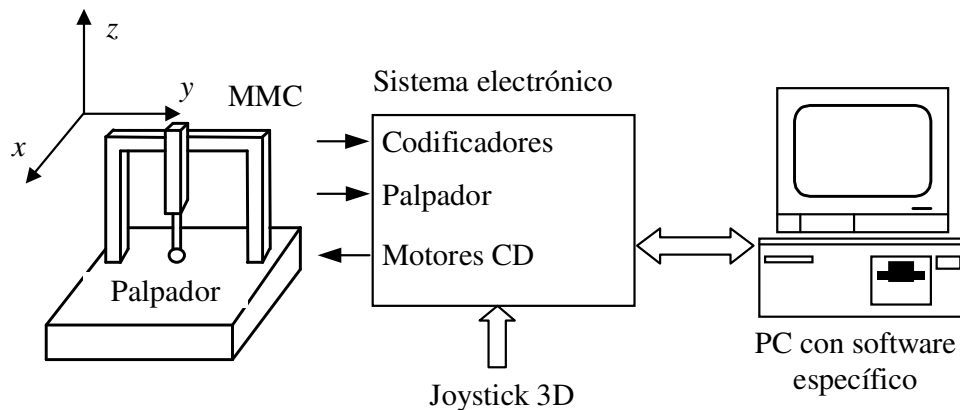
En el CCADET UNAM, particularmente en el Laboratorio de Metrología, se ha desarrollado una MMC, con el objetivo de brindar tecnología acorde a las necesidades particulares de medición geométrica tri-dimensional a las cuales se enfrenta esta organización.

Las MMC's son uno de los instrumentos de medición geométrica de mayor precisión y exactitud que existen en el área de la metrología dimensional. Una MMC determina posiciones espaciales al posicionar su dispositivo sensor, llamado palpador de contacto, en los ejes coordenados. La posición espacial es registrada por transductores de longitud, comúnmente codificadores ópticos, en el momento en que se registra un contacto mecánico entre el palpador y la pieza de trabajo. En este sentido, una MMC incorpora los siguientes elementos que le proporcionan un elevado nivel de desempeño:

- Un diseño mecánico con tres ejes coordenados de alta precisión que permiten el desplazamiento micrométrico del dispositivo sensor o palpador de contacto. El diseño mecánico varía en función de las aplicaciones, pero generalmente es un prototipo robusto, del tipo conocido como "puente" y permite la medición en volúmenes desde fracciones de metro hasta varios metros.
- Un palpador de contacto con cinco niveles de palpación: dos en el eje  $x$ , dos en el eje  $y$  y uno en el eje  $z$  [20, 21].
- Un sistema mecánico de transmisión de movimiento para transformar el movimiento angular de tres motores de Corriente Directa, CD, a los tres desplazamientos lineales de los ejes coordenados.
- Un sistema electrónico de control realimentado de posición y velocidad para tres motores CD.
- Tres codificadores ópticos lineales solidarios a los ejes coordenados para el registro de posición.
- Un joystick 3D o palanca de mandos para que el operador pueda controlar los tres movimientos de cada uno de los ejes coordenados de forma intuitiva.

- Una computadora que desarrolle los algoritmos de control y medición así como la interfase de operación con el usuario.
- Software de medición para el ajuste de posiciones espaciales a geometrías típicas.

La figura 1 muestra el esquema general de una MMC con los elementos descritos.



**Figura 1.** Esquema general de una MMC.

### Antecedentes

El prototipo desarrollado en el CCADET UNAM constituye un instrumento de medición, el cual ha sido utilizado para el desarrollo de algoritmos de medición y hardware asociado, en donde participan académicos y alumnos del laboratorio. La gran complejidad de este tipo de instrumentos de medición y el reducido tamaño de la organización nos ha obligado a abordar el proyecto en etapas. En el año de 1999 se concluyó la primera etapa del prototipo mecánico que incluía un sistema rudimentario para el registro de las lecturas ( $x$ ,  $y$ ,  $z$ ), mediante una consola electrónica de despliegue. En esta situación, la MMC desarrollada hasta ese momento era capaz de medir manualmente con alta exactitud, pero sin las capacidades de control automático y software de medición, figura 2.

En una segunda etapa, y con el auge y disponibilidad en los sistemas digitales, en el año de 2001 se incluyó un sistema electrónico y software para el registro de lecturas y ajuste a geometrías típicas sobre la base de circuitos electrónicos comerciales, software desarrollado en Visual C++ y una PC, figura 3 [1, 2]. Las capacidades de la MMC en esa fecha permitían medir en forma semi-automática con gran exactitud, software de medición e interfase a una PC. Sin embargo, el posicionamiento se realizaba de forma manual, ya que carecía de las capacidades de control automático.





**Figura 2.** MMC del CCADET en su primera etapa.



**Figura 3.** MMC del CCADET en su segunda etapa.

En una tercera etapa, en el año 2004 se experimentó con hardware y software para el control de movimiento en motores CD [22]. Los experimentos se realizaron en un prototipo de MMC de un solo eje coordinado, figura 4 [23]. En este desarrollo se prescindió de la MMC, pero los resultados proporcionaron la pauta para extender el sistema electrónico y software utilizados a los tres ejes coordinados.



**Figura 4.** MMC del CCADET en su tercera etapa.

### **Descripción del problema a resolver**

Dados los antecedentes descritos con anterioridad y el avance paulatino por etapas del proyecto, nos han conducido hacia el desarrollo de un sistema de control para tres servomotores con aplicación a la transmisión de movimiento en la MMC del CCADET UNAM. Entonces el problema consiste en desarrollar hardware y software para controlar, en lazo cerrado, la velocidad y posición de tres motores CD de forma completamente automática, centralizada en una PC y ofreciendo las capacidades de comodidad y amigabilidad que se ofertan en las MMC comerciales. Por lo tanto, se contempla la manipulación de dispositivos tales como codificadores ópticos y motores CD de mediana potencia. En forma adicional, el sistema a desarrollar deberá contemplar la interfase con un palpador mecánico de contacto y un joystick 3D. Las pruebas del proyecto a desarrollar se realizarán sobre un prototipo de MMC ya que se carece del diseño mecánico para la transmisión de movimiento en la MMC del CCADET.

### **Relevancia y justificación**

La relevancia del presente proyecto de tesis radica en el uso de recursos de bajo costo, que potencialmente nos conduzcan a ofrecer un instrumento adecuado a las necesidades nacionales, así como apoyo a la creación de infraestructura de desarrollo propio que puede ser utilizada en la prestación de servicios ofrecidos por el Laboratorio de Metrología o que puede ser ofertada como una transferencia tecnológica. Adicionalmente, el trabajo apoya el desarrollo de un proyecto ambicioso en donde se pretende construir un instrumento de medición de alta exactitud y gran capacidad de automatización. En

dicho proyecto participan alumnos y académicos realizando labores de investigación aplicada.

### **Alcance y limitaciones**

Las complicaciones implícitas al desarrollo de un proyecto orientado a automatizar un instrumento de gran exactitud, limitan nuestro trabajo a un prototipo de sistema para la automatización del movimiento en una MMC. La automatización completa implica no sólo el desarrollo electrónico y de computación propuesto, sino también el diseño de mecanismos de precisión que posicionen a la MMC en los niveles de exactitud deseados. De esta manera, nos limitamos a desarrollar un prototipo electrónico y de programación que sea fácilmente transportado a nuestra MMC como parte de una etapa posterior de integración de partes.

Por otro lado, la infraestructura a utilizar ya ha sido adquirida y por lo tanto no existe posibilidad de evaluación de diferentes opciones debido a que la experiencia de los académicos del Laboratorio de Metrología ha orientado el proyecto a su situación actual.

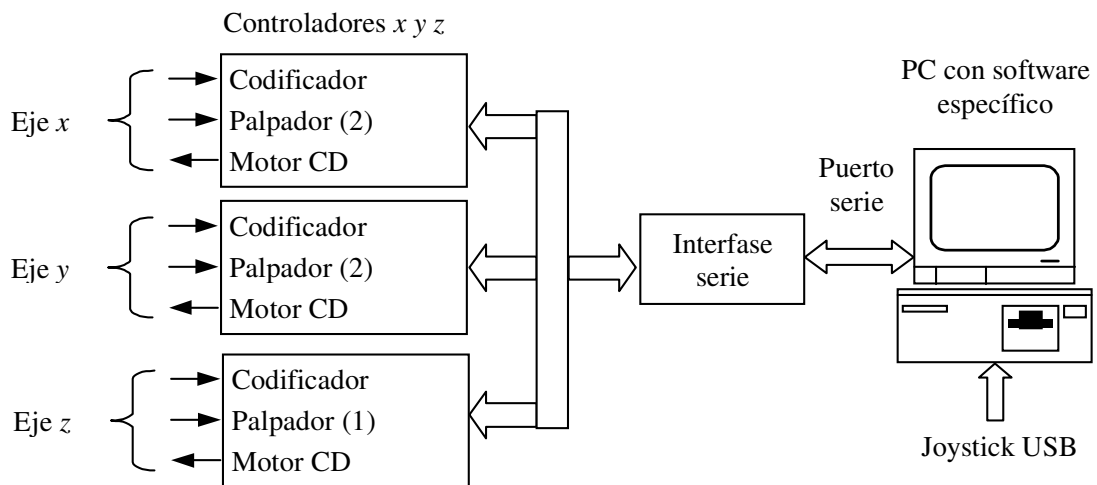
### **Relación con otras áreas**

Dado que el prototipo desarrollado es muy ambicioso, existe una gran relación con otras áreas, tales como control digital, robótica, programación de computadoras, sistemas electrónicos digitales, medición e instrumentación y metrología dimensional.

## **Método**

El método a emplear en el presente proyecto consiste en integrar un sistema de control de movimiento realimentado orientado a una MMC, empleando técnicas de control digital sobre la base de una PC, tarjetas electrónicas comerciales y una PC. La figura 5 muestra el esquema propuesto.

Debido a la carencia actual de un diseño mecánico de precisión que nos permita transmitir el movimiento de los motores CD a los ejes coordenados en la MMC, la implementación del esquema de la figura 5 se limita a desarrollar un prototipo que simule las condiciones reales. La validación del prototipo nos conducirá a la implementación en una MMC real.



**Figura 5.** Esquema del sistema de control para tres servomotores con interfase a una PC.

## Resultados esperados

Desarrollar un sistema hardware y software para el control de tres servomotores gobernado por una PC que sea fácilmente integrado a una MMC. Se espera que tenga una exactitud en el posicionamiento de  $\pm 2$  cuentas del codificador óptico, interfase con un palpador de contacto con 5 niveles de palpación e interfase con joystick 3D.

## Resumen de la tesis

El presente documento se ha dividido en tres grandes capítulos y los cuales se describen brevemente a continuación.

En el primero de ellos, veremos un pequeño esbozo histórico de este tipo de máquinas en el ámbito mundial. Conoceremos también lo que es y lo que hace una MMC; las partes que la constituyen, los conceptos fundamentales y cómo lleva a cabo su funcionamiento.

En el segundo capítulo, se presenta un esquema general del sistema electromecánico. Se explicará de manera más amplia y detallada cada uno de los dispositivos, circuitos y herramientas utilizadas en dicho sistema. Esto es, encontraremos la descripción del hardware y el software desarrollados para realizar el control de los servomotores.

Finalmente, en el tercer y último capítulo, se analizan los resultados y las conclusiones a las que se llegaron, una vez terminado y puesto a prueba el prototipo de la MMC. Para avalar esto último, el sistema tuvo que pasar por dos pruebas experimentales diseñadas para verificar la confiabilidad en las mediciones. De esta manera, se podrán

corroborar dichos resultados de manera gráfica y confrontarlos con los objetivos inicialmente planteados. Asimismo, se anexa un apartado en el que se describe el trabajo a futuro o por realizar, una vez que se han obtenido tales resultados.

Se ha incluido en la parte final, un apéndice con el objeto de documentar el trabajo adicional que se tuvo que llevar a cabo para la realización de este proyecto.

# CAPÍTULO 1

## ANTECEDENTES

En los procesos de control de calidad se manifiesta constantemente la necesidad de sistemas de evaluación más acertada de las características geométricas individuales o de subconjuntos de los productos industriales. Entre la diversidad de medios implicados en encontrar solución al problema mencionado, la MMC es el medio más complejo y eficaz.

A finales del año 1962, la firma italiana *DEA* construyó la primera máquina de medición en un garaje de Borgo San Paolo, cerca de Turín [16].

En 1973 *Zeiss* creó la máquina UMM500, equipada con un palpador universal, un ordenador y un control numérico (CN) [3]. En esta composición se encuentran por primera vez los dos elementos fundamentales que definen una máquina de medición tridimensional moderna: el *palpador universal* y el *ordenador*. El palpador universal es un captador de posición que, mediante contacto, permite localizar puntos sobre cualquier tipo de superficie. La figura 1.1 muestra una MMC comercial del tipo conocido como “puente”.



**Figura 1.1.** MMC comercial tipo puente marca Zeiss.

## 1.1. Máquinas de Medir por Coordenadas

La Máquina de Medición por Coordenadas (MMC) puede ser definida como una máquina que emplea tres componentes móviles que se trasladan a lo largo de guías con recorridos ortogonales, para medir una pieza por determinación de las coordenadas  $x$ ,  $y$  y  $z$  de los puntos de la misma con un palpador de contacto o sin él y sistema de medición del desplazamiento (escala), que se encuentran en cada uno de los ejes. Como las mediciones están representadas en el sistema tridimensional, la MMC puede efectuar diferentes tipos de medición como: dimensional, posicional, desviaciones geométricas y mediciones de contorno [12].

Las MMC's son instrumentos que sirven para realizar mediciones dimensionales y de desviaciones de la regularidad geométrica de objetos con forma simple o compleja. Las hay de distintas dimensiones, tipos, materiales y exactitudes de medición y para aplicación en laboratorios de metrología, laboratorios industriales y en las líneas de producción.

La gran exactitud en la medición dimensional de las MMC es posible gracias a la diversidad de tecnologías que la componen, entre las cuales podemos mencionar:

- Mecánica de precisión. Un diseño mecánico de gran precisión y robustez que permite el posicionamiento exacto y la medición en los tres ejes coordenados.
- Control automático. Sistemas de control de movimiento en lazo cerrado que proporcionan al instrumento la capacidad de determinar posiciones con una gran precisión.
- Software de medición. Algoritmos de medición que permiten ajustar series de puntos a geometrías dimensionales con gran precisión.
- Transductores<sup>1</sup>. Dispositivos sensores de longitud y contacto que habilitan a la máquina para realizar mediciones sobre piezas físicas minimizando la intervención por parte del operador y la formación de incertidumbre.

Aunque las MMC son diferentes entre sí, dependiendo del volumen de medición y la aplicación para las que son fabricadas, todas operan bajo el mismo principio: el registro de una pieza con una técnica de medición punto a punto, asignando a cada uno de éstos una terna de coordenadas referido a un sistema coordinado en 3D; y la vinculación numérica de las coordenadas asignadas a los puntos, con una geometría espacial completa de la pieza a través de un software de medición en un equipo de procesamiento de datos. Sin embargo, el modo en que el software es concebido es propio para cada máquina y dependiendo del diseñador y programador éste será capaz de manejar y parametrizar apropiadamente los elementos geométricos regulares que normalmente son manejados en una MMC, lo cual define prácticamente lo que la MMC puede medir.

Los procedimientos de medición y procesamiento de datos de las MMC, poseen una serie de características que se describen a continuación: primeramente se tiene un sistema de posicionamiento que provoca que el palpador alcance cualquier posición en  $x$ ,  $y$  o  $z$ ; este sistema de posicionamiento es accionado a través de unos motores, que a su vez, poseen unos codificadores ópticos, los que producirán una señal adecuada para activar un contador que incrementan su número con relación a la posición del eje con respecto de su origen.

---

<sup>1</sup> Un transductor es un dispositivo que convierte un tipo de energía determinada en señales eléctricas (corriente o voltaje).



En este sistema, como en muchos otros, es de primordial importancia la existencia de un origen para poder determinar la posición.

El sistema dispondrá además de un palpador que, al ser accionado, hará que los datos del contador del sistema de posicionamiento sean trabajados por la unidad principal de la MMC y sean transformados en coordenadas  $x$ ,  $y$  y  $z$  y además se apliquen las fórmulas programadas para después desplegar los datos en una pantalla.

La máquina también posee una palanca de control que acciona directamente los servomotores provocando un desplazamiento manual de cada uno de los ejes.

Esta MMC en particular posee teclado para introducción de datos, un monitor que proporciona la visualización de ellos, ya sea que se introduzcan o se generen por la MMC.

Como se mencionó anteriormente, el palpador que se encuentra en el extremo inferior del eje Z, se acciona al toque de la pieza que se desea medir. En la figura 1.2 podemos ver representada de manera esquemática y sencilla a una MMC.

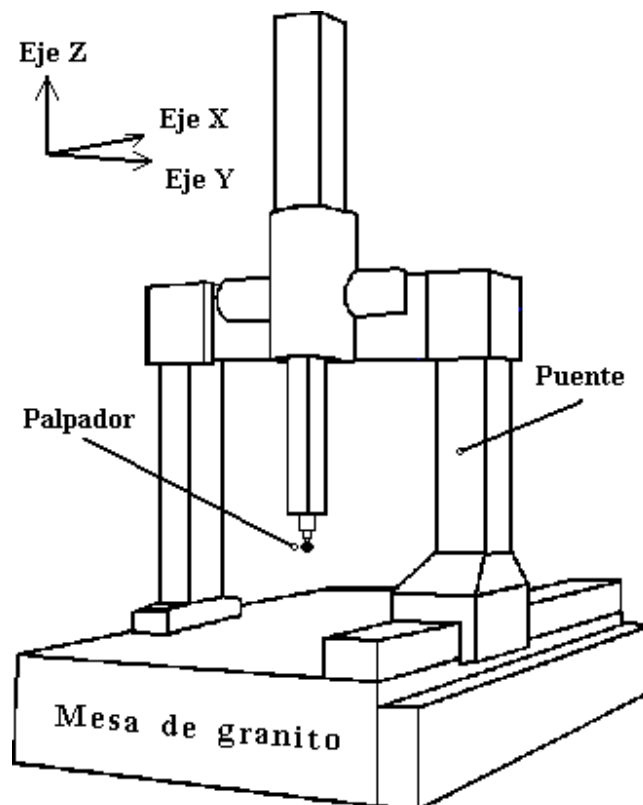


Figura 1.2. Esquema simple de una MMC.

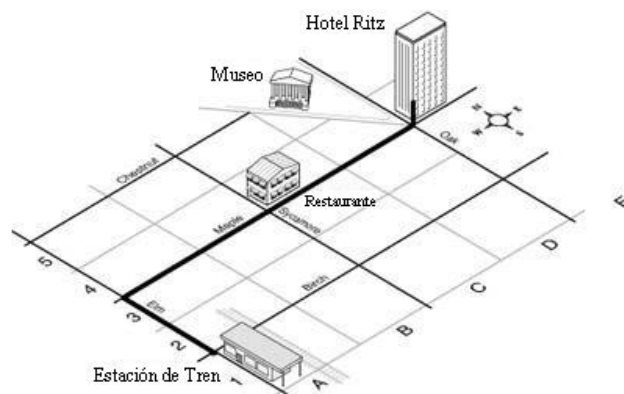
### 1.1.1. Entendiendo la MMC

#### Sistema de Coordenadas

Utilizamos *el sistema de coordenadas* para describir los movimientos de una máquina de medición. El famoso sistema por coordenadas, inventado por el conocido filósofo y matemático René Descartes a principios del siglo XVII, nos permite localizar características en relación con otras características de las piezas.

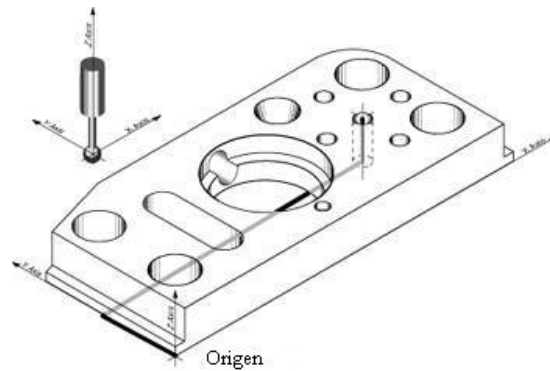
El sistema por coordenadas se parece mucho a un plano de alzado en el que la combinación de una letra a un extremo del plano con un número al otro extremo, y alzados a lo largo del plano, es lo que describe cada localización en el mapa. Esa combinación letra/número/alzado es lo que se conoce como una coordenada y representa un lugar concreto en relación con los otros.

Un ejemplo de esto, lo podemos apreciar mediante el mapa de una calle con los edificios representados, figura 1.3. Para ir a su habitación del hotel Ritz desde la estación de tren (su origen), debe andar 2 manzanas de Elm Street, 4 manzanas de Marple y subir 3 plantas del Ritz. Esa misma localización se puede describir con las coordenadas 4-E-3 del mapa, que corresponderían a los ejes  $x$ ,  $y$  y  $z$  de la máquina. Esas coordenadas sólo describen la localización de su habitación en el mapa y no otras localizaciones.



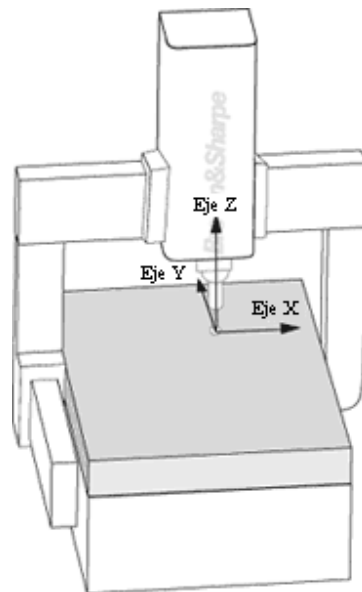
**Figura 1.3.** Mapa.

Una MMC trabaja casi de la misma forma que su dedo cuando éste traza coordenadas en un mapa; los tres ejes de la máquina. En lugar de un dedo, la máquina utiliza un sensor para medir puntos en la pieza, figura 1.4. Cada punto de la pieza es único para el sistema de coordenadas de la máquina. La MMC combina los puntos medidos para formar una característica que se pueda relacionar con todas las otras características.



**Figura 1.4.** Sensor utilizado por una MMC.

Hay dos tipos de sistemas de coordenadas en el mundo de la medición. El primero se conoce como *Sistema de coordenadas para la máquina*. Aquí los ejes  $x$ ,  $y$ , y  $z$  (figura 1.5) se refieren a los movimientos de la máquina. Si se toma una perspectiva desde la parte frontal de la máquina, el eje  $x$  va de izquierda a derecha, el eje  $y$  va hacia adelante y hacia atrás y el eje  $z$  va de arriba a abajo, verticalmente perpendicular a los otros dos ejes.

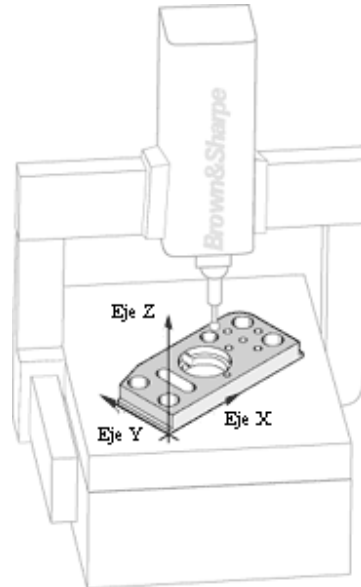


**Figura 1.5.** Sistema de coordenadas para la máquina.

El segundo sistema de coordenadas se conoce como *Sistema de coordenadas para la pieza*. Los tres ejes tienen relación con los datos o características de la pieza, figura 1.6.

Antes de la introducción de software para llevar a cabo mediciones por coordenadas, las piezas se alineaban de forma paralela a los ejes de la máquina de manera que los sistemas de coordenadas para la máquina y la pieza estaban paralelos el uno con el otro. Esto requería

mucho tiempo y los resultados no eran muy precisos. Cuando la pieza era redondeada o contorneada, en lugar de cuadrangular o triangular, la tarea de medición era casi imposible.



**Figura 1.6.** Medición de una pieza.

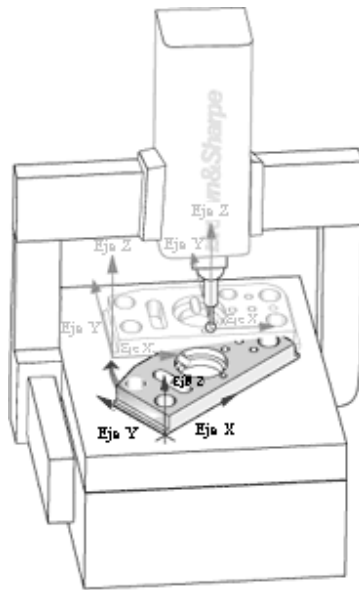
Con el software de las MMC de hoy, se miden los datos de las piezas (a partir de la impresión de la parte), establecen el Sistema de Coordenadas de la Pieza y matemáticamente lo ponen en relación con el Sistema de Coordenadas de la Máquina.

### **Alineamiento**

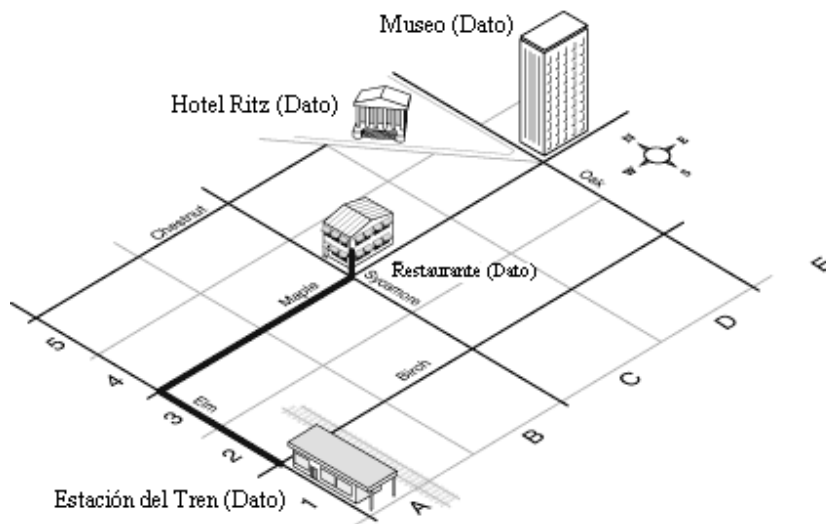
El proceso que se lleva a cabo para relacionar los dos sistemas de coordenadas se llama *alineamiento*, figura 1.7. Con un mapa de calles, lo hacemos automáticamente girando el mapa de forma que sea paralela a una calle (dato) o a la dirección de una brújula (el Norte). Cuando hacemos eso, lo que estamos haciendo en realidad es situarnos a nosotros mismos en el “sistema de coordenadas del mundo”.

### **Dato**

Un *dato* es una localización. Podemos usar los datos como guías para decir a los otros dónde nos encontramos o como señas para llegar a un sitio, figura 1.8. En el mapa el Hotel Ritz es un dato, también lo son la estación del tren, el museo y el restaurante. Así, si utilizamos un origen, datos, señas y distancias, la gente tendrá toda la información que necesita para ir de una localización a otra.



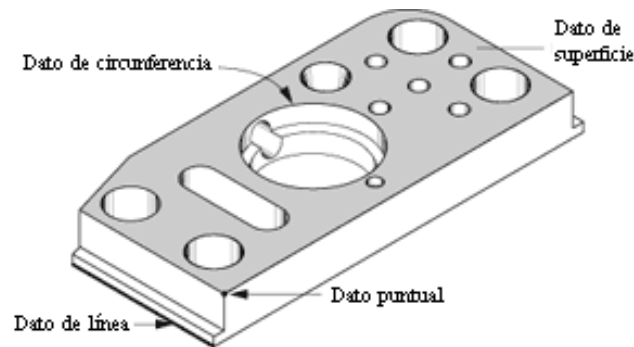
**Figura 1.7.** Alineamiento.



**Figura 1.8.** Localizaciones en un Mapa.

Por ejemplo, para ir de la estación de tren (origen) al restaurante, debe caminar dos manzanas de Elm Street hacia el norte (dato), girar a la derecha y moverse dos manzanas de Maple hacia el este (dato).

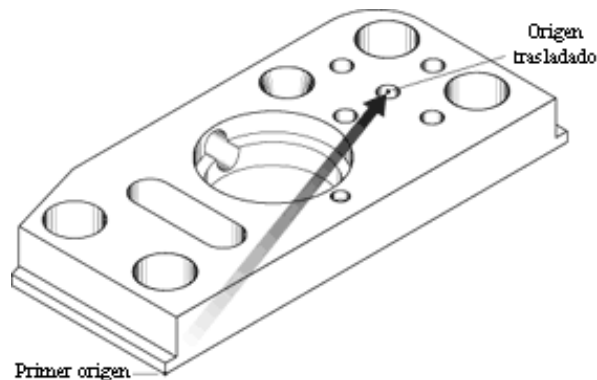
En metrología, un dato es una característica de una pieza, como por ejemplo un agujero, una superficie o una ranura. Medimos una pieza para determinar la distancia entre una característica y otra, figura 1.9.



**Figura 1.9.** Pieza con características especiales.

### Traducción

Imagínese que necesita saber la distancia entre una característica de la pieza y otra, figura 1.10. Queremos saber, por ejemplo, la distancia que hay entre un agujero central y los centros de cuatro agujeros. Para hacerlo, tendremos que medir primero el agujero central, *traduciremos* el *origen* en el centro de ese agujero y después mediremos cada uno de los agujeros de alrededor. El hecho de mover el punto inicial (*origen*) de la medición a otra parte de la pieza es lo que se conoce como *traducción*.



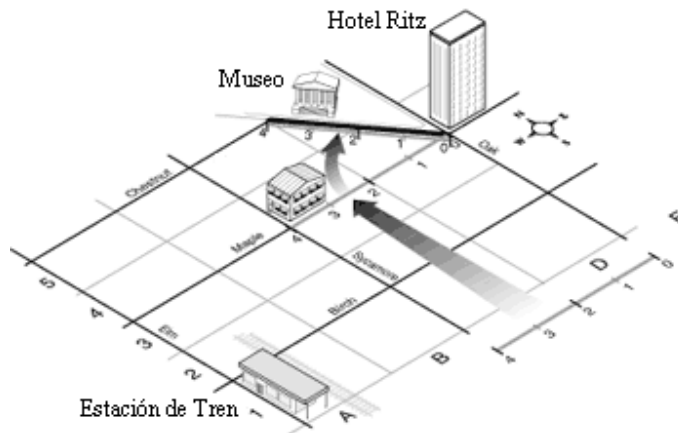
**Figura 1.10.** Traducción.

En nuestro mapa de la calle, supongamos que una vez que llega a su hotel, decide comer en un restaurante muy conocido. Primero deberá encontrarlo en el mapa. Ahora, el hotel se convierte en su punto inicial u *origen*. Si sabe dónde se encuentra, mirando el mapa verá que deberá andar dos manzanas al oeste por Maple Street hasta llegar al restaurante.

### Rotación

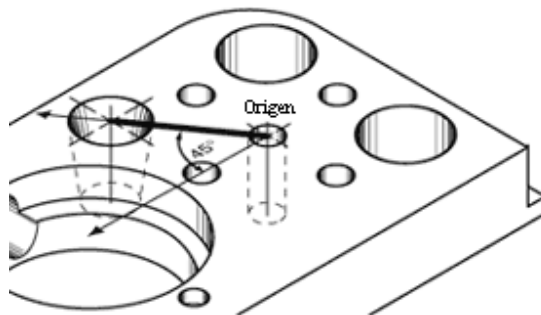
No todos los datos forman un ángulo recto con otros datos. Por ejemplo, si nos fijamos en nuestro plano de calles (figura 1.11), vemos que el museo está situado en una calle que no es ni paralela ni forma

un ángulo recto con las calles del hotel, del restaurante o de la estación del tren. Así, para determinar la distancia que hay entre el hotel y el museo, primero tendrá que *traducir* su *punto de origen* al hotel y entonces *rotar* el punto inicial de manera que esté paralelo a la calle del museo. Ahora puede medir fácilmente la distancia entre el museo y el hotel.



**Figura 1.11.** Rotación.

El mismo proceso se le aplica a la pieza, figura 1.12. La distancia entre los dos agujeros de la pieza se puede medir una vez que el origen se *traduce* al agujero más pequeño y el *sistema por coordenadas de la pieza* se rota de forma automática  $45^\circ$ . Ahora los dos agujeros están alineados en el eje de Y y la distancia se puede medir automáticamente.



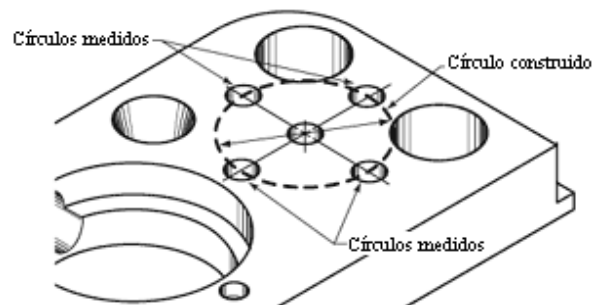
**Figura 1.12.** Rotación en una pieza.

### Características medidas y construidas

La gran mayoría de piezas están formadas por elementos geométricos simples hechos a máquina o por moldeo. Estos elementos primarios (superficies planas, bordes, cilindros, esferas, conos, etc.) se conocen como características. Cuando una MMC puede medir esas características de forma directa palpando las superficies

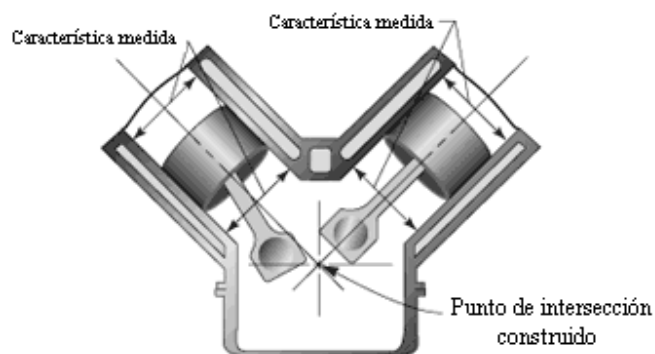
que forman la característica con un sensor, las características son *características medidas*.

Otras características como la distancia, la simetría, la intersección, el ángulo y la proyección no se pueden medir directamente, pero se tienen que construir de forma matemática a partir de características ya medidas, antes de que se puedan determinar sus valores. Esas son las *características construidas*. En la figura 1.13, la línea central del círculo se construye desde los puntos centrales de los 4 círculos ya medidos.



**Figura 1.13.** Características en una pieza.

Las relaciones entre una o varias características y otra característica o grupo de características son cruciales para la fabricación. Por ejemplo, el punto de intersección entre los cilindros a un lado del bloque de un motor y los del otro lado es lo que determina si las piezas aparejadas encajan bien, figura 1.14. Este punto de intersección se construye a partir de dos características ya medidas (los cilindros del motor).



**Figura 1.14.** Medición de un motor.

### Compensación volumétrica

Aunque la avanzada tecnología de fabricación permite tolerar y hacer piezas de forma muy precisa, aún existen imperfecciones. Por pequeñas que sean, que existan tolerancias significa que existen errores.



Las MMC no son diferentes a otros productos en este aspecto. Aunque están hechas con tolerancias muy ajustadas, hay errores (de rotación, inclinación, giradas, rectitud, ortogonalidad y de escala) en su estructura que afectan a su eficacia. Cuanto más ajustadas sean las tolerancias, más precisas tendrán que ser las MMC.

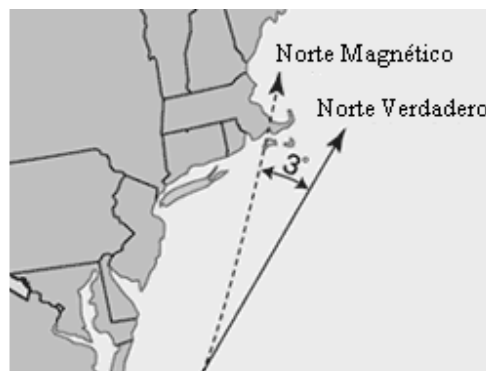
La mayoría de las imprecisiones de las MMC se pueden corregir automáticamente en el ordenador de la misma. Una vez medidos todos los errores geométricos de la MMC (llamados *errores cartográficos*) se pueden minimizar o incluso eliminar con potentes algoritmos del software de la máquina. Ese proceso se conoce como *compensación del error volumétrico*.

Con la eliminación matemática de errores se reduce el costo de fabricación.

La *compensación volumétrica* se puede entender como la relación entre un mapa y una brújula. Si usted quiere navegar hasta un lugar concreto, tendrá que saber la dirección exacta que debe seguir desde su posición (origen). Utilizaremos un mapa y una brújula para determinar la dirección a tomar. Sin embargo, hay una diferencia entre el norte verdadero y el norte magnético. La diferencia entre los dos es la *variación* y es la causa de la inexistencia de uniformidad en el campo magnético terrestre. Así, para determinar la dirección correcta desde un punto a otro, a partir de lo que marque la brújula se le tendrá que añadir o sustraer la variación entre el norte y el norte magnético.

En el mapa que se presenta en la figura 1.15, la diferencia entre el norte real y el norte magnético ( $3^\circ$  W) se tiene que compensar, si no el navegante acabaría yendo al noroeste y tocaría tierra antes de llegar al destino final.

Una MMC realiza automáticamente una compensación similar para eliminar de la medición las variaciones de la máquina.



**Figura 1.15.** Compensación.

## Compensación de los sensores

Por lo general, las MMC toman los datos palpando la pieza con un sensor adjunto en los ejes de medición de la máquina. El sensor puede ser sólido o electrónico y funcionar con un accionador. Aunque el palpador del sensor es muy preciso, una vez que el sensor se ha adjuntado a la MMC, el posicionamiento del palpador en el sistema de la máquina de medición por coordenadas se tiene que determinar antes de medir. Puesto que es la circunferencia la que palpa la pieza, el centro y el radio del sensor se determinan midiendo una esfera muy precisa (*esfera de recalificación*).

Cuando se conocen el radio y el centro del palpador, cuando el sensor toma contacto con una pieza, el radio del palpador compensa de forma automática las coordenadas del palpador según el punto real de contacto del palpador, figura 1.16. La dirección de la compensación se determina automáticamente por el procedimiento de alineación.

Hacemos un procedimiento similar cuando aparcamos un coche. Cuanto mejor podamos calcular la compensación del exterior del coche, más cerca del bordillo aparcaremos.

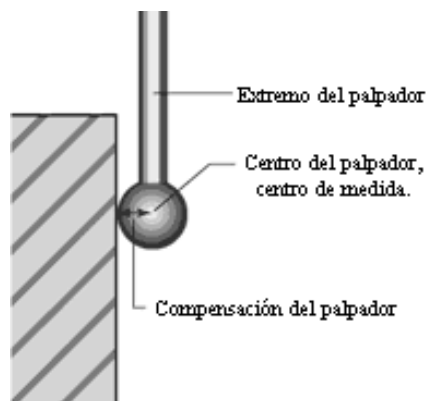


Figura 1.16. Compensación en el palpador.

## Proyecciones

Una *proyección* es la reproducción de la característica de una pieza a otra característica, como por ejemplo proyectar un círculo o una línea en un plano o un punto en una línea.

La proyección de la característica de una pieza en otra se puede comparar con la creación del tradicional mapamundi. El mapamundi se consigue proyectando un mapa del mundo (esfera) encima de un cilindro, figura 1.17.

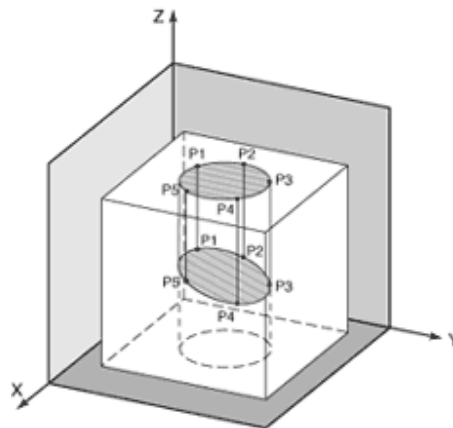
En metrología, las *proyecciones* le permiten medir de forma más precisa cómo encajarán finalmente dos piezas que tienen que ir juntas.

Para realizar mediciones de cilindros automovilísticos (ej. bloques de motor), puede determinar de forma precisa cómo encajarán los pistones en el cilindro y cómo encajará la cámara de combustión en la cabeza proyectando un cilindro en el plano de la cara de la cabeza.



**Figura 1.17.** Proyección de un mapamundi.

Se necesita un mínimo de tres puntos para medir el diámetro de un círculo y si los puntos no están a la misma distancia desde la parte superior del soporte, el diámetro se presentará en forma elíptica, figura 1.18. Para vencer esta mala interpretación, los datos de la medición se proyectan en un plano que está perpendicular a la línea central del cilindro. El resultado es una determinación precisa del tamaño real de la característica de la pieza.

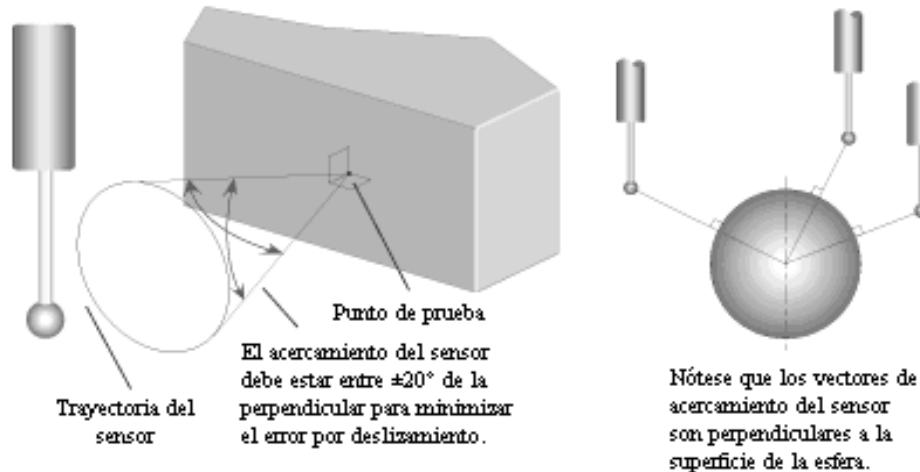


**Figura 1.18.** Representación en forma elíptica de un círculo.

### 1.1.2. Técnicas efectivas del sensor

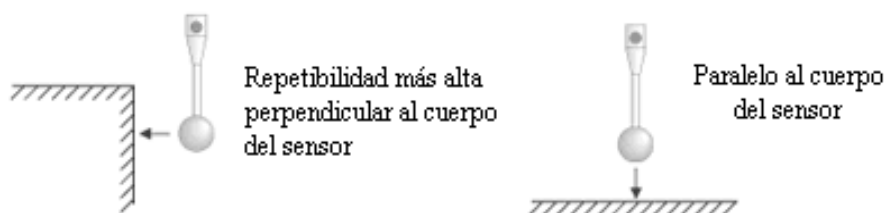
Con el uso de técnicas efectivas del sensor, se pueden eliminar muchas causas comunes de errores de medición al inspeccionar una pieza.

Por ejemplo, el sensor tendría que tomar las medidas perpendiculares a la superficie de la pieza siempre que sea posible, figura 1.19. Los sensores con accionador están diseñados para dar unos resultados óptimos cuando el palpador del sensor toca la pieza perpendicular al cuerpo del sensor. Sería ideal si pudiese realizar tomas dentro de unos  $20^\circ$  de perpendicular para evitar que resbale el palpador del sensor. Los resbalones producen resultados inconsecuentes y que no se pueden repetir.



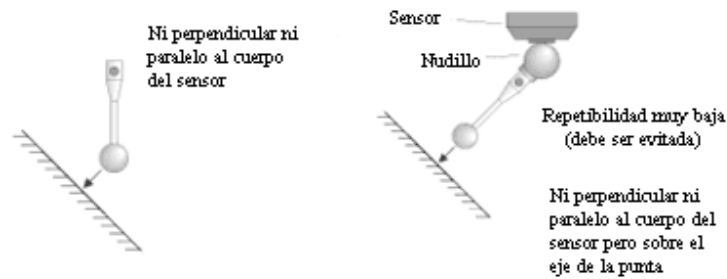
**Figura 1.19.** Errores comunes al inspeccionar piezas.

Las tomas del sensor paralelas al cuerpo del mismo, en otras palabras, a lo largo del eje de la aguja, no tienen tanta repetibilidad como las tomas perpendiculares al eje, figura 1.20.



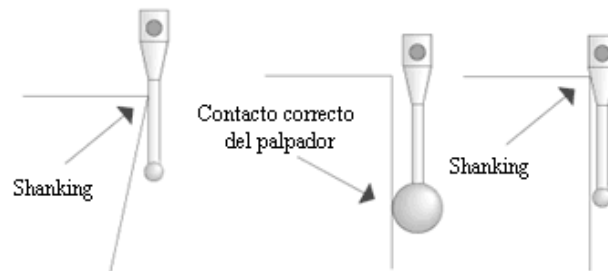
**Figura 1.20.** Tomas paralelas y/o perpendiculares del sensor.

Las tomas del sensor que no sean ni perpendiculares ni paralelas al cuerpo del mismo, ofrecerán resultados que tienen incluso menos repetibilidad que las tomas paralelas al cuerpo del sensor, figura 1.21. Debería evitarse realizar tomas del sensor paralelas a la aguja y a un ángulo con el cuerpo del sensor, puesto que producirían grandes errores.



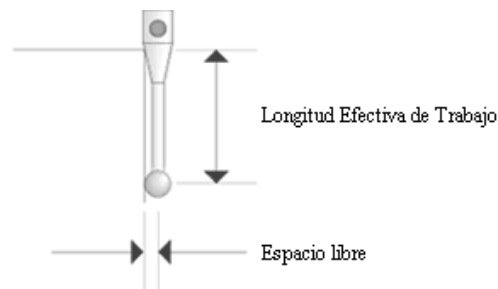
**Figura 1.21.** Tomas no perpendiculares ni paralelas del sensor.

Cuando el asta de la aguja toca la pieza (*shanking*), también se pueden producir errores de medición, figura 1.22. Cuando el sensor toma contacto con la pieza con el asta de la aguja y no con el palpador, el sistema de medición considerará que la toma se ha realizado de manera normal y eso conllevará a más errores.



**Figura 1.22.** Error típico de medición.

Uno puede reducir las posibilidades de *shanking* utilizando un palpador de diámetro mayor para aumentar la distancia entre bola/asta y la superficie de la pieza. Por lo general, cuanto mayor sea el diámetro del palpador, la aguja podrá recorrer distancias más profundas antes de tocar la pieza. Eso se llama *longitud efectiva* del sensor (figura 1.23). Del mismo modo, cuanto mayor sea el palpador, menor efecto tendrá en la superficie acabada de la pieza, ya que el punto de contacto abarca una zona mayor de la característica que está siendo medida. Sin embargo, la medida de los palpadores se limita al tamaño del agujero más pequeño a ser medido.



**Figura 1.23.** Longitud efectiva del sensor.

Los resultados de la medición de puntos tomados con un sensor electrónico quedan grabados cuando la aguja se desvía lo suficiente para romper contactos mecánicos o para generar la fuerza suficiente como para accionar el sistema de circuitos eléctricos de presión-sensibilidad. La configuración física de los contactos causa pequeños errores de precisión, aunque se pueden reducir durante la calificación del sensor. Sin embargo, cuanto mayor sea la extensión del palpador, mayor será el error del predesplazamiento y mayor será el error residual que se queda después de la calificación del sensor. Los sensores largos no son tan rígidos como los cortos. Cuanto más se doble o se desvíe la aguja, menor será la precisión. Debería evitarse el uso de sensores con combinaciones de aguja/extensión muy largas.

### 1.1.3. Parametrización de elementos geométricos en una MMC

Los softwares comerciales básicos de MMC cuando menos, manejan los elementos geométricos regulares como son el punto, la línea, el plano, el círculo, la esfera, el cilindro, el cono y normalmente las representan de la siguiente manera:

- El **punto**, que es el elemento más simple, es adecuadamente parametrizado en una MMC por una terna de coordenadas. Por ejemplo,  $(x, y, z)$ .
- La **línea** es manejada y parametrizada adecuadamente a través de cualquiera de las siguientes expresiones:
  - Un punto sobre la línea  $(x_0, y_0, z_0)$  y los cosenos directores que gobiernan la dirección de la línea  $(a, b, c)$ , tal que cualquier punto sobre la línea  $(x, y, z)$  satisfaga la ecuación (1.1) para algún valor del parámetro  $t$ .
 
$$(x, y, z) = (x_0, y_0, z_0) + t(a, b, c) \quad (1.1)$$
  - Dos puntos sobre la línea  $(x_1, y_1, z_1)$ ,  $(x_2, y_2, z_2)$ , tal que cualquier punto sobre la línea  $(x, y, z)$  satisfaga la ecuación (1.2) para algún valor del parámetro  $t$ .
 
$$(x, y, z) = (1-t)(x_1, y_1, z_1) + t(x_2, y_2, z_2) \quad (1.2)$$
- El **plano** es manejado y parametrizado adecuadamente a través de cualquiera de las siguientes expresiones:
  - Un punto sobre el plano  $(x_0, y_0, z_0)$  y los cosenos directores  $(a, b, c)$  de la normal al plano tal que cualquier punto en el plano  $(x, y, z)$  satisfaga la ecuación (1.3).
 
$$a(x - x_0) + b(y - y_0) + c(z - z_0) = 0 \quad (1.3)$$

- Un punto sobre el plano  $(x_0, y_0, z_0)$  y un punto sobre la normal al plano  $(x_1, y_1, z_1)$  tal que cualquier punto en el plano  $(x, y, z)$  satisfaga la ecuación (1.4).

$$(x_1-x_0)(x-x_0) + (y_1-y_0)(y-y_0) + (z_1-z_0)(z-z_0) = 0 \quad (1.4)$$

- El **círculo** es representado mediante las coordenadas del centro, el diámetro o radio y con las características del plano sobre el cual recae, sin embargo, en algunos softwares sólo es manejado a través del centro y el radio y no da información acerca del plano sobre el cual está contenido, pero es posible conocer éste a través de algunos alineamientos básicos.
- La **esfera** es uno de los elementos que para la mayoría de softwares de medición de MMC es parametrizada correctamente, pero manejada incorrectamente como un punto que corresponde al centro de la esfera; y el diámetro o radio de la misma y no como una superficie.
- El **cilindro**, al igual que la esfera, no es manejado correctamente, puesto que para la mayoría de los softwares de MMC es representado con las características de una línea que es el eje y su radio o diámetro y no como una superficie.
- El **cono**, del mismo modo que el cilindro, es manejado con las características de dirección de una línea, el ángulo del cono y las coordenadas del ápice, o bien la distancia de la superficie del cono a un punto sobre el eje del cono.

El software también debe ser capaz de realizar cálculos y aplicaciones geométricamente válidas como: la obtención de distancias, intersecciones, ángulos y creación de líneas y superficies auxiliares entre elementos geométricos, siempre de acuerdo a la manera en que son representados los elementos geométricos de medición para la MMC y es ahí en donde empiezan las limitaciones de estos.

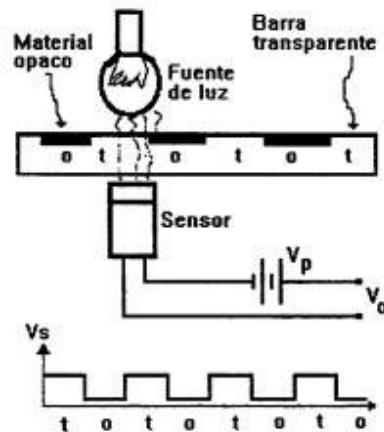
## 1.2. Codificadores ópticos

Existen los codificadores lineales y angulares. Estos son transductores que producen directamente una salida digital cuando miden desplazamiento lineales o angulares [15]. Por su codificación directa, no hay necesidad de usar convertidores análogo digital (A-D), ya que las señales eléctricas generadas son en forma digital.

Estas señales pueden ser acumuladas por un contador o microprocesador (sistema incremental) o simplemente leídas en un

puerto periférico para su procesamiento. La transducción de movimiento puede realizarse usando tres mecanismos de codificación diferentes: codificación por escobillas, codificación magnética, codificación óptica o fotoeléctrica.

En nuestro caso, nos interesa la codificación óptica, la cual se ejemplifica en la figura 1.24



**Figura 1.24.** Esquema de un codificador.

En ésta se tiene una barra o disco translucido con incrustaciones de material opaco o viceversa. De esta forma y usando un emisor y un detector de luz infrarroja, se generan pulsos a la salida de un fototransistor que ha sentido el movimiento de la barra o disco. Cabe mencionar que muchas veces el material del disco es opaco, y que simplemente se le maquinan hoyos simétricamente espaciados para lograr el mismo efecto.

Por su confiabilidad y simplicidad, los codificadores ópticos son dominantes en la industria. Existen 2 tipos de codificadores ópticos: los incrementales y los absolutos.

Los codificadores ópticos incrementales son aquellos que producen pulsos igualmente espaciados y que pueden ser acumulados por un contador (para incrementar o decrementar). El conteo es la indicación del desplazamiento. El punto de origen puede fijarse o programarse en el dispositivo de lectura.

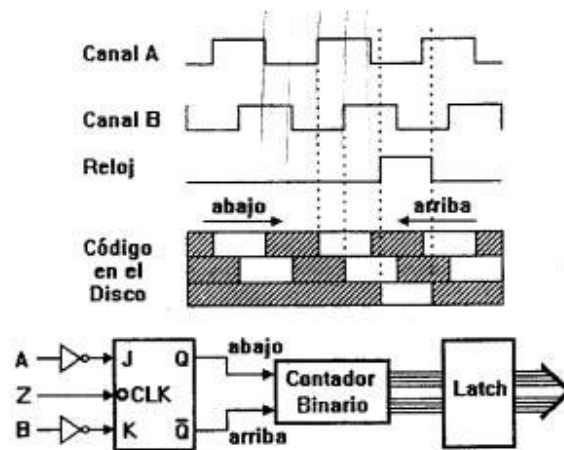
La mayoría de los sistemas codificadores incrementales tienen 2 salidas (canales) en cuadratura para verificar o determinar la dirección del movimiento. Esto permite contar las transiciones y observar el estado del otro canal durante las transiciones para determinar si el canal A adelanta al B y entonces derivar la dirección.



La importancia de esta información de dirección se deriva debido a la vibración inherente a cualquier sistema mecánico. Un error en la cuenta ocurrirá si el codificador, usando un solo canal (tipo tacómetro) se detiene durante una transición. Si las fuerzas de vibración producen un rebote en la señal codificada (transición), el contador seguirá contando cada transición aún cuando el sistema aparezca parado.

Usando un detector de cuadratura en un codificador de dos canales y viendo la transición en su relación al estado del canal opuesto, se puede generar la información confiable de dirección.

La figura 1.25 muestra un decodificador de cuadratura usando un circuito J-K y 2 canales desfasados  $\frac{1}{4}$  de pulso para lograr la detección de dirección.



**Figura 1.25.** Decodificador de cuadratura.

Los codificadores absolutos son aquellos que generan un número codificado digitalmente, y es indicativo de la posición. Esto se hace mediante un conjunto de cabezas lectoras y una disposición de segmentos en multicanal. Los códigos digitales típicos son: binario natural, gray y BCD (decimal codificado en binario).

Estos códigos se ilustran en la tabla 1.1.

Binario natural	Gray	BCD
0000	0000	0000 0000
0001	0001	0000 0001
0010	0011	0000 0010
0011	0010	0000 0011
0100	0110	0000 0100
0101	0111	0000 0101
0110	0101	0000 0110
0111	0100	0000 0111
1000	1100	0000 1000

1001	1101	0000 1001
1010	1111	0001 0000
1011	1110	0001 0001
1100	1010	0001 0010
1101	1011	0001 0011
1110	1001	0001 0100
1111	1000	0001 0101

**Tabla 1.1.** Códigos para codificadores absolutos.

### 1.3. Control de movimiento en motores CD

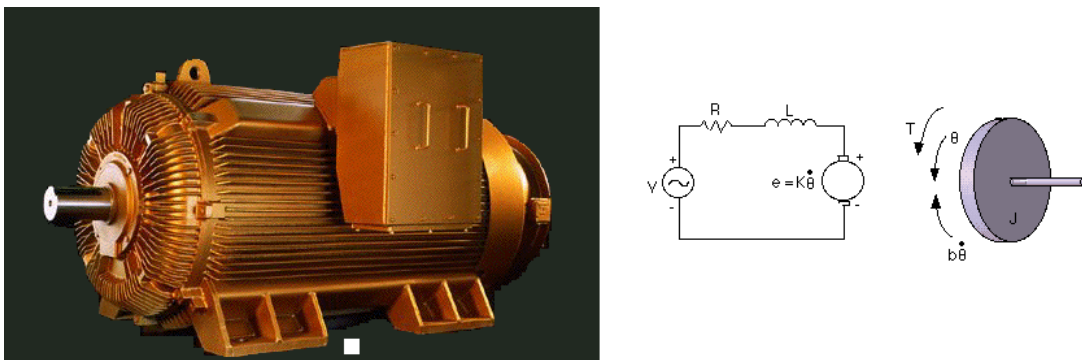
Para llevar a cabo un control de movimiento preciso en una MMC, es necesario un buen sistema que permita dicho control. Existen en el mercado una gran diversidad de fabricantes de controladores de motores, ya sean motores a pasos, motores CA, o CD según las necesidades de los usuarios.

Para este proyecto, se ha usado una tarjeta de control PIC-SERVO v.4 que está constituida por un módulo controlador, un amplificador, una interfase para codificador óptico incremental, interfase para interruptores de límite, una entrada con pre-amplificador auxiliar y una interfase de comunicaciones que se adapta al puerto serie de la computadora.

El sistema construido se mostrará en detalle y se explicará más a fondo en el capítulo siguiente.

#### 1.3.1. Modelado de motores CD

Un actuador común en los sistemas de control es el motor CD. Este provee directamente movimiento rotatorio y, junto con las ruedas, tambores y cables, también puede proporcionar movimiento transitorio. El motor, el circuito eléctrico y el diagrama de cuerpo libre del rotor, se muestran en la figura 1.26.



**Figura 1.26.** Modelado de un motor CD.

Como ejemplo, asumiremos los siguientes valores para los parámetros físicos:

- Momento de inercia del rotor  $(J) = 0.01 \text{kg} \left( \frac{\text{m}^2}{\text{s}^2} \right)$
- Cociente de humedad del sistema mecánico  $(b) = 0.1 \text{Nms}$
- Constante de la fuerza electromotriz  $(K = K_e = K_t) = 0.01 \frac{\text{Nm}}{\text{A}}$ 
  - Resistencia eléctrica  $(R) = 1\Omega$
  - Inductancia eléctrica  $(L) = 0.5\text{H}$
  - Entrada  $(V)$  : voltaje de la fuente
  - Salida  $(\theta)$ : posición del eje
- Se asume que el rotor y el eje son rígidos

El torque del motor,  $T$ , se relaciona a la corriente de armadura,  $i$ , por un factor constante  $K_t$ . La fuerza electromotriz,  $e$ , que regresa, se relaciona a la velocidad rotacional a través de las ecuaciones 1.5 y 1.6.

$$T = K_t i \quad (1.5)$$

$$e = K_e \dot{\theta} \quad (1.6)$$

En unidades del Sistema Internacional<sup>2</sup> (el cual usaremos),  $K_t$  (constante de armadura) es igual a  $K_e$  (constante del motor).

De la figura 1.26, podemos escribir las ecuaciones 1.7 y 1.8 con base en las leyes de Newton y las de Kirchhoff.

$$J \ddot{\theta} + b \dot{\theta} = K i \quad (1.7)$$

$$L \frac{di}{dt} + Ri = V - K \dot{\theta} \quad (1.8)$$

### Función de transferencia

Usando las transformadas de Laplace, las ecuaciones modeladas 1.7 y 1.8 pueden ser expresadas en términos de  $s$ .

---

<sup>2</sup> El Sistema Internacional de Unidades se fundamenta en siete unidades de base correspondientes a las magnitudes de longitud, masa, tiempo, corriente eléctrica, temperatura, cantidad de materia, e intensidad luminosa. Estas unidades son conocidas como el **metro**, el **kilogramo**, el **segundo**, el **ampere**, el **kelvin**, el **mol** y la **candela**, respectivamente. A partir de estas siete unidades de base, se establecen las demás unidades de uso práctico, conocidas como unidades derivadas, asociadas a magnitudes tales como velocidad, aceleración, fuerza, presión, energía, tensión, resistencia eléctrica, etc [3].

$$s(Js + b)\Theta(s) = KI(s) \quad (1.9)$$

$$(Ls + R)I(s) = V - Ks\Theta(s) \quad (1.10)$$

Eliminando  $I(s)$ , podemos obtener la siguiente función de transferencia en lazo abierto, donde la velocidad rotacional es la salida y el voltaje es la entrada.

$$\left(\frac{\theta}{V}\right) = \frac{K}{(Js + b)(Ls + R) + K^2} \quad (1.11)$$

### Espacio-Estado

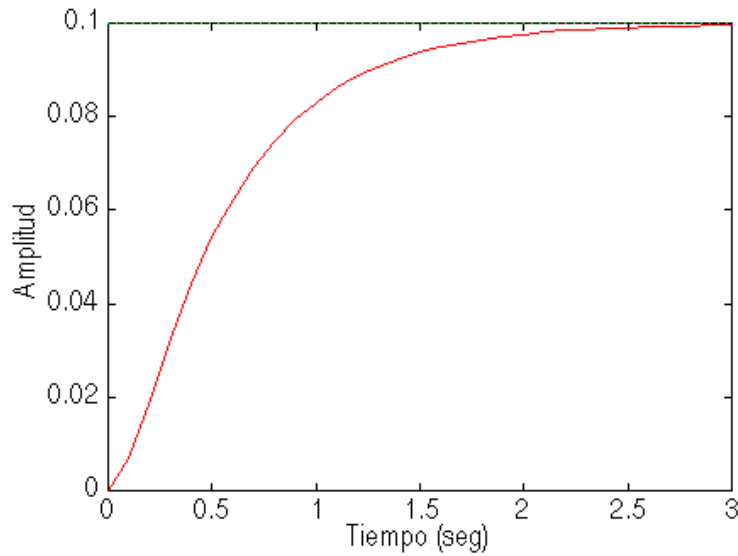
En la forma espacio-estado, las ecuaciones anteriores pueden ser expresadas eligiendo una velocidad rotacional y una corriente eléctrica como las variables de estado y el voltaje como entrada. La velocidad rotacional es elegida como la salida:

$$\frac{d}{dt} \begin{bmatrix} \theta \\ i \end{bmatrix} = \begin{bmatrix} -\frac{b}{J} & \frac{K}{J} \\ -\frac{K}{L} & -\frac{R}{L} \end{bmatrix} \begin{bmatrix} \dot{\theta} \\ i \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{L} \end{bmatrix} V \quad (1.12)$$

$$\dot{\theta} = [1 \ 0] \begin{bmatrix} \dot{\theta} \\ i \end{bmatrix} \quad (1.13)$$

### Requisitos de diseño

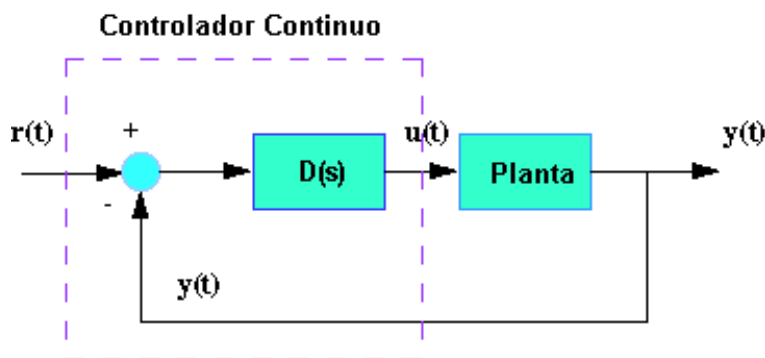
Primeramente, el motor sin compensar puede rotar únicamente a razón de 0.1 rad/seg con un voltaje de entrada de 1 V. Puesto que el requisito básico de un motor es que debe de girar a la velocidad deseada, el error de estado estacionario de la velocidad debe ser menor al 1%. El otro requisito de diseño es que el motor debe de girar hasta llegar a la velocidad de estado estacionario tan pronto como es puesto en funcionamiento. En este ejemplo, se requiere un tiempo de asentamiento de 2 segundos. Dado que una velocidad que supere la de referencia puede dañar al equipo, se elige que el sobrepaso sea menor al 5%. Si simulamos dichas condiciones para ver cómo responde nuestro sistema al escalón, obtendríamos una gráfica como se muestra en la figura 1.27.



**Figura 1.27.** Respuesta al escalón para el sistema en lazo abierto.

### 1.3.2. Control digital

La figura 1.28 muestra el típico sistema continuo realimentado. La mayoría de los controladores continuos pueden ser construidos usando electrónica analógica.



**Figura 1.28.** Diagrama de bloques de un controlador continuo.

El controlador continuo, mostrado en el cuadro punteado, puede ser remplazado por un controlador digital, figura 1.29, ya que realiza la misma tarea. La única diferencia entre ambos controladores radica en que el sistema digital opera con señales discretas (o muestras de la señal detectada) en vez de señales continuas.

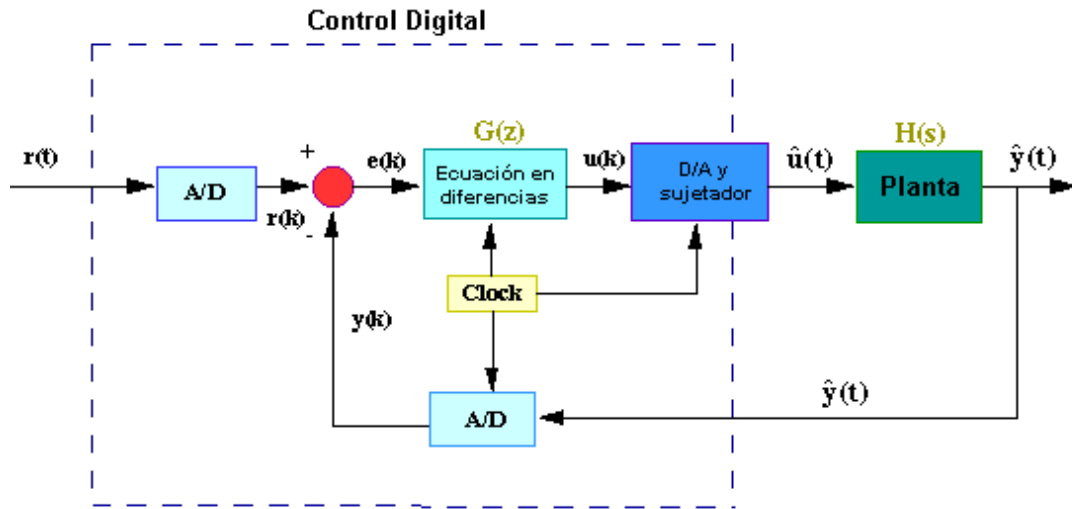


Figura 1.29. Control digital.

Diversos tipos de señales del diagrama esquemático anterior, pueden ser representados por los diagramas de la figura 1.30.

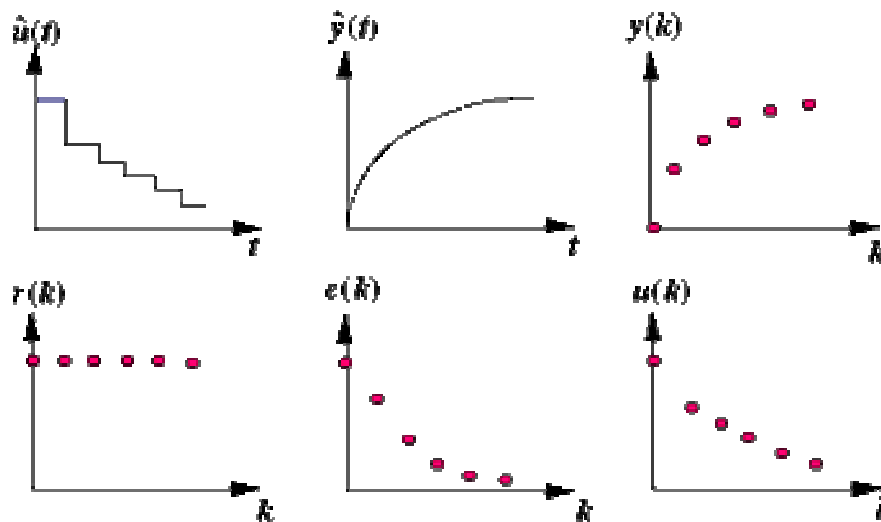
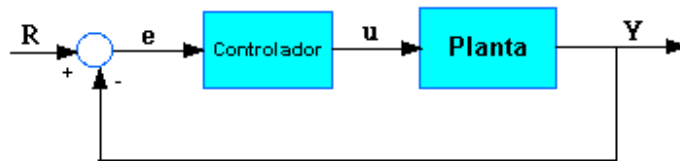


Figura 1.30. Señales.

### 1.3.3. Control PID

Para explicar lo que es un controlador PID, supongamos que tenemos el siguiente diagrama de la figura 1.31.



**Figura 1.31** Control PID.

donde: la planta es el sistema a ser controlado y el controlador proporciona la excitación a la planta y está diseñado para tener el control total sobre el comportamiento del sistema.

La función de transferencia de un controlador PID se describe como:

$$K_p + \frac{k_I}{s} + K_D s = \frac{K_D s^2 + K_p s + K_I}{s} \quad (1.14)$$

$K_p$  = Ganancia proporcional.

$K_I$  = Ganancia integral.

$K_D$  = Ganancia derivativa.

Primero veamos cómo funciona un controlador PID en un sistema de lazo cerrado usando la figura 1.31. La variable  $e$  representa el error, esto es, la diferencia entre el valor deseado de entrada ( $R$ ) y la salida actual ( $Y$ ). La señal de error es enviada al controlador PID, quien analiza ambas, la derivativa y la integral de dicha señal de error. La señal  $u$  que se encuentra entre ambos bloques, es ahora igual a la ganancia proporcional ( $K_p$ ) veces la magnitud del error más la ganancia integral ( $K_I$ ) veces la integral del error más la ganancia derivativa ( $K_D$ ) veces la derivada del error, o bien:

$$u = K_p e + K_I \int e dt + K_D \frac{de}{dt} \quad (1.15)$$

La señal  $u$  es enviada hacia la planta y es obtenida una nueva señal de salida ( $Y$ ). Esta señal ( $Y$ ) es mandada de nueva cuenta de regreso al sensor para obtener una nueva señal de error  $e$ . El controlador toma esta nueva señal de error y computa su ganancia derivativa e integral. El proceso se realiza una y otra vez.

### Características de los controladores P, I y D

Un controlador proporcional ( $K_p$ ) tendrá el efecto de reducir el tiempo de subida y reducirá, más nunca eliminará, el error de estado estacionario. El control integral ( $K_I$ ) eliminará el error de estado

estacionario pero producirá una muy mala respuesta transitoria. Finalmente, el control derivativo ( $K_D$ ) incrementará la estabilidad del sistema, reduciendo el sobrepaso y mejorando la respuesta transitoria. Cada uno de los efectos producidos por los controladores P, I y D, en lazo cerrado, se resumen en la tabla 1.2.

Respuesta	Tiempo de subida	Tiempo de Sobrepaso	Tiempo de asentamiento	Error de estado estacionario
$K_P$	Decrementa	Incrementa	Cambio mínimo	Decrementa
$K_I$	Decrementa	Incrementa	Incrementa	Lo elimina
$K_D$	Cambio mínimo	Decrementa	Decrementa	Cambio mínimo

**Tabla 1.2** Resumen de efectos de cada controlador.

Cabe señalar que estas correlaciones pueden no ser muy exactas, ya que los controladores  $K_P$ ,  $K_I$ , y  $K_D$  son dependientes uno del otro. De hecho, el modificar una de estas variables, cambia el efecto de las otras dos. Por esta razón, la tabla se debe utilizar únicamente como referencia cuando se requieran determinar dichos valores.

### Consejos generales para diseñar un control PID

Cuando se diseña un control PID para un sistema dado, es importante tomar en cuenta los siguientes pasos para obtener la respuesta deseada:

- 1) Obtener la respuesta en lazo abierto y determinar lo que se necesite mejorar.
- 2) Agregar un control proporcional para mejorar el tiempo de subida.
- 3) Agregar un control derivativo para mejorar el sobrepaso.
- 4) Agregar un control integral para eliminar el error de estado estacionario.
- 5) Ajustar cada uno de los controles  $K_P$ ,  $K_I$ , y  $K_D$  la respuesta general deseada.

Finalmente, cabe agregar que no siempre es prioritario agregar un control de este tipo a un sistema. Eso dependerá de la respuesta a obtener. Por ejemplo, si una vez que se prueba que un control PI es suficiente para entregar la respuesta deseada, con este bastará para trabajar. No olvidar que entre más sencillo sea un controlador, más fácil será su uso y aplicación [5].



## Servomotor

Un servomotor es un motor eléctrico con los engranajes de reducción adecuados para obtener un pequeño movimiento con mucha fuerza. Además incorpora la electrónica de control para posicionarlo con precisión en una posición determinada, esto es, tenemos la capacidad de controlarlo en velocidad y/o posición. Dicho motor emplea retroalimentación para proveer señales precisas de arranque y detención.

- Control en velocidad: posibilidad de hacer girar al motor a una velocidad determinada independientemente de la carga o fuerza que deba vencer.
- Control en posición: posibilidad de realizar desplazamientos entre dos posiciones determinadas con precisión.

Debemos tener muy en cuenta que, cuando se trabaja con servomotores hay que cuidar ciertos detalles, como los siguientes:

- 1) Poner mucha atención en las **TIERRAS**. La tierra (cable negro) del servo tiene que ir conectada a la tierra de su fuente de alimentación, esto es, a la salida de cero volts, así como a la tierra del microcontrolador, computadora o lo que se esté usando para controlar.
- 2) Si se usan cables demasiado largos para controlar los servos, es muy probable que se tenga ruido (tartamudeo) en dichos servos. Esto ocurre porque mientras más largo es el cable, resulta más vulnerable a ruido electromagnético (efecto antena), e incluso es perturbado por señales de otros servos. Esto se soluciona utilizando un cable blindado (coaxial), recordando obviamente aterrizar el blindaje.
- 3) Tratar de no cargar demasiado peso a los servos. Un servo en operación normal **NO** se debe de calentar. Si se calienta es que se le está pidiendo que sostenga más peso del que sea capaz de soportar y por ende acorte su vida útil. Hay que recordar que un servo se creó para mover algo, no para cargarlo.
- 4) Siempre que sea posible, utilizar fuentes de voltaje separadas para cada servo y para la electrónica digital. Cuando se controla al servo mediante una PC, no habrá que preocuparse por esto, ya que ésta tiene su fuente propia. Pero si se requiere manejar al servo por medio de un controlador, es muy recomendable trabajar con fuentes independientes con las tierras compartidas porque los servomotores generan bastante ruido hacia su línea de alimentación.

- 5) Los servos envejecen con el uso. Si se les trata bien, pueden durar algunos años funcionando correctamente. Si el servo comienza a tartamudear y se está seguro que no es por ninguna de las causas anteriores, todavía se puede tratar de recalibrarlo. Esto significa cambiar el intervalo de tiempo entre los pulsos de control hasta encontrar el más óptimo. Otra opción es desarmarlo y limpiar el potenciómetro que tienen en su interior con algún spray limpiador. Si todo lo anterior falla y el servo definitivamente ya no logra funcionar bien, aún se puede desarmar y reutilizar el motor con su caja de engranes, y a veces incluso aprovechar parte de la electrónica de control de su tarjeta para convertirlo en un motor bidireccional [6].

# **CAPÍTULO 2**

## **SISTEMA PROPUESTO**

En este capítulo se explicará de manera más amplia y detallada cada uno de los dispositivos, circuitos y herramientas utilizadas para la puesta en marcha del proyecto.

### **2.1. Esquema general**

Una Máquina de Medición por Coordenadas (MMC), se opera mediante un programa que ha sido desarrollado específicamente para esta aplicación, implementando la interfase de usuario, así como el soporte de los dispositivos periféricos que complementan el sistema y cuyo objeto es el de tener la mínima intervención por parte del usuario, automatizando la mayor parte del mecanismo y facilitando de esta manera, su uso. Dicho programa se detallará más adelante.

Los aspectos de interés en el control de movimiento de la MMC son:

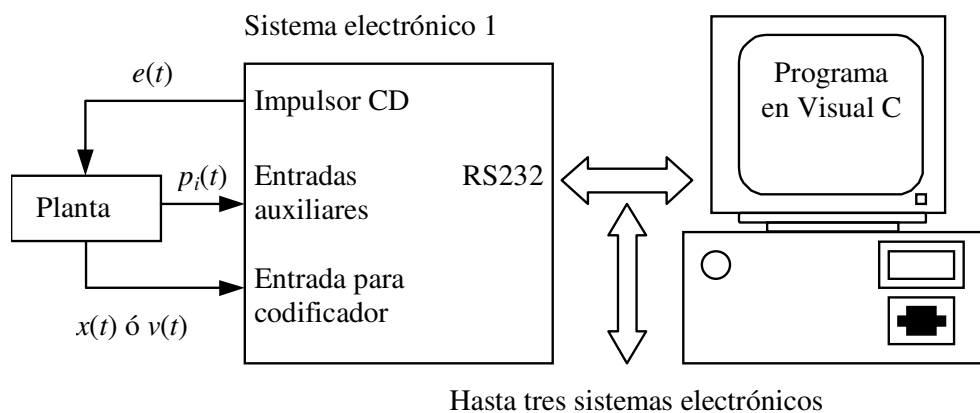
- Procesamiento en tiempo real de algoritmos PID<sup>3</sup> para el control de posición y velocidad en motores CD usando perfiles de aceleración preferidos.
- Soporte para la programación de algoritmos específicos en la generación de perfiles de movimiento y ajuste de parámetros.
- Compatibilidad con transductores de posición del tipo codificador incremental lineal.

---

<sup>3</sup> Proporción-Integración-Derivación. Se utiliza como comparador para la velocidad y posición del motor.

- Entradas y salidas digitales de propósito general para capturar las posiciones de disparo que genera un palpador de contacto.
- Soporte para el control de uno hasta 3 ejes coordenados.

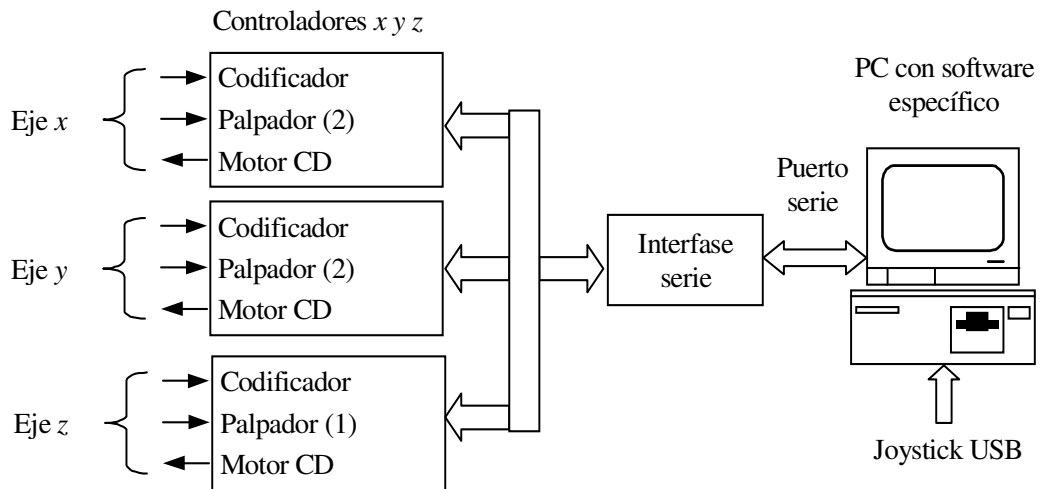
De manera esquemática, podemos describir el sistema propuesto. En la figura 2.1 podemos observar la planta, que representa uno de los tres ejes coordenados a controlar. El sistema electrónico realimentado que impulsa el motor CD implementa un algoritmo PID que genera tensiones de alimentación,  $e(t)$  y procesa posiciones,  $x(t)$ , o velocidades,  $v(t)$  del codificador óptico para cerrar el lazo de control. Las entradas auxiliares,  $p_i(t)$   $i=1, 2, \dots, 5$  son utilizadas para capturar las señales de disparo que genera el palpador de contacto. Una computadora gobierna hasta tres sistemas electrónicos como el de la figura mediante un único enlace de comunicaciones RS232<sup>4</sup>. En éste esquema, la computadora conforma un control maestro y los sistemas electrónicos constituyen controles esclavos. El programa en Visual C++ 6 gobierna los algoritmos de control de movimiento y delega a los sistemas electrónicos los algoritmos de control PID de posición o velocidad. El software en la PC proporciona parámetros que representan trayectorias de movimiento y el sistema electrónico las ejecuta.



**Figura 2.1.** Esquema del sistema propuesto para el control de movimiento en una MMC para un solo eje.

Ahora bien, una vez realizados los trabajos propios para un eje, se realiza lo mismo para los siguientes dos. La figura 2.2 ilustra de manera esquemática el modelo propuesto para los tres ejes.

<sup>4</sup> Es el método de comunicaciones más común. Las dos primeras siglas (RS) significan Recommended Standard o Estándar Recomendado.



**Figura 2.2.** Esquema completo del sistema propuesto para el prototipo de la MMC.

En la figura 2.2, los tres circuitos electrónicos controladores reciben las señales en cuadratura que generan tres codificadores ópticos lineales solidarios a cada uno de los ejes coordenados en una supuesta MMC. Las señales en cuadratura son procesadas de manera que sea interpretada la lectura de longitud ( $x$ ,  $y$ ,  $z$ ). El palpador en la MMC genera dos señales eléctricas on/off cuando hace contacto con una pieza de trabajo para los ejes  $x$  y  $y$  y una señal para el eje  $z$ . Tales señales son recibidas por los controladores en sus entradas para interruptores con el fin de validar una lectura de longitud en el momento preciso en que ocurre el contacto mecánico. Cada controlador implementa un algoritmo de control digital PID para el control de la planta conformada por el conjunto de motor CD, eje coordinado mecánico y codificador óptico. Una PC con software desarrollado específicamente para la aplicación, envía y recibe comandos de cada controlador. Los controladores comparten una sola interfase de comunicación serial. Lo anterior se logra al multiplexar en el tiempo mediante un protocolo de comunicación que direcciona los módulos controladores. Los comandos son ingresados por el operador en una interfase de medición amigable. En particular se requiere de un control de palanca o joystick 3D que tradicionalmente es usado en las MMC's.

## 2.2. Descripción hardware

A continuación se dará una breve descripción acerca de cada uno de los componentes y dispositivos usados para el desarrollo del prototipo.

### 2.2.1. Circuitos y módulos para el control de motores CD

Para lograr una fiabilidad y precisión mayores en una máquina MMC, es necesario tener un buen control para el movimiento de los motores. Esto se logra a través de controladores digitales formados básicamente por circuitos electrónicos [18]. Existen en el mercado muchos fabricantes y por ende una enorme diversidad de controladores para motores CD, CA, motores a pasos, etc.

En nuestro caso, se optó por un sistema de control de motores PIC-SERVO, que está constituido por un módulo controlador y un adaptador para puerto serie RS232 [10, 11]. El fabricante es JRKERR<sup>5</sup>. La pregunta obligada es: ¿por qué elegir a este fabricante? Las respuestas son simples. Las tarjetas son adecuadas para los requerimientos que necesitábamos, así como el precio y la facilidad de interacción entre los módulos de control con la PC a través del puerto serie.

### 2.2.2. Módulo de control

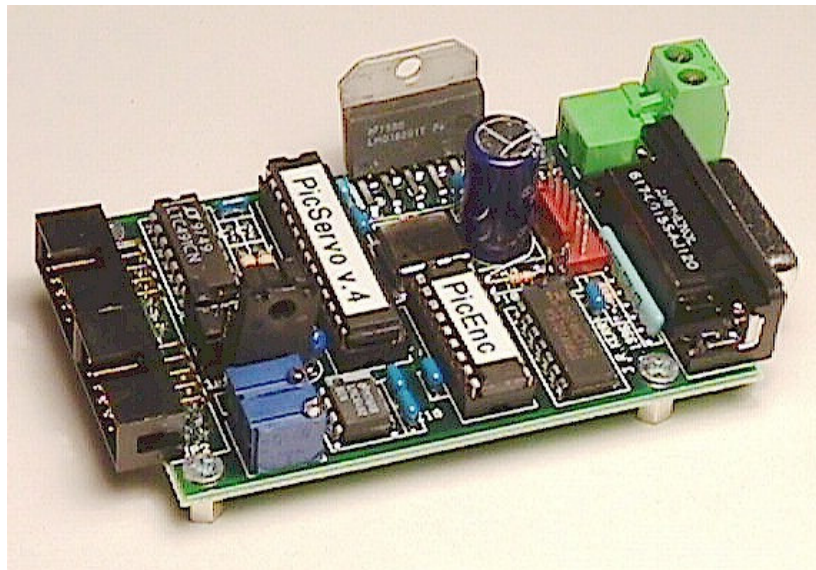
La tarjeta de control PIC-SERVO KAE-T0V4-DPS es un sistema completo de servo-control que incluye un servo controlador, un amplificador, una interfase de comunicaciones, una interfase para codificador óptico incremental, interfase para interruptores de límite y una entrada con pre-amplificador auxiliar, figura 2.3. Esta tarjeta está diseñada para soportar hasta 32 controladores que pueden ser conectados directamente a un sólo puerto serie estándar (con un adaptador para puerto serie RS232 ó RS485). En nuestra propuesta para el control de movimiento de la MMC presentada en la figura 2.1, el módulo de control es conectado de la siguiente forma:

- La salida del amplificador energiza el motor CD,  $e(t)$ .
- La interfase para codificador manipula un codificador óptico,  $x(t)$  o  $v(t)$ .
- La interfase para interruptores límite se conecta a las señales de disparo en el palpador,  $p_i(t)$ .

Los dispositivos del PIC-SERVO KAE-T0V4-DPS forman el núcleo principal del módulo. El PIC-SERVO es un microcontrolador PIC16C73 programado con un controlador PID y una interfase de comandos seriales. El PIC-ENC es un microcontrolador PIC16C54 programado como un codificador incremental de 16 bits. Ambos dispositivos están basados en la serie de microcontroladores PIC16CXX.

---

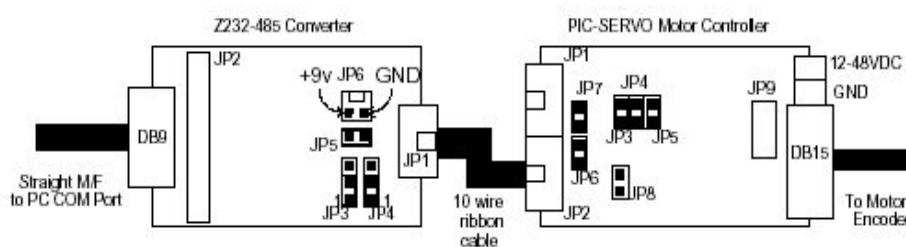
<sup>5</sup> <http://www.jrkerr.com>



**Figura 2.3.** PIC-SERVO KAE-T0V4-DPS.

La conexión básica para un sólo controlador que proporciona la funcionalidad deseada mostrada en la figura 2.1, es presentada en la figura 2.4. Se muestra un módulo de control conectado hacia una computadora mediante un módulo adaptador para puerto serie. A su vez, el módulo de control se conecta a un motor CD, al codificador incremental y a los interruptores en el palpador a través del conector DB15<sup>6</sup>. El sistema de la figura 2.4 requiere de los siguientes dispositivos de soporte.

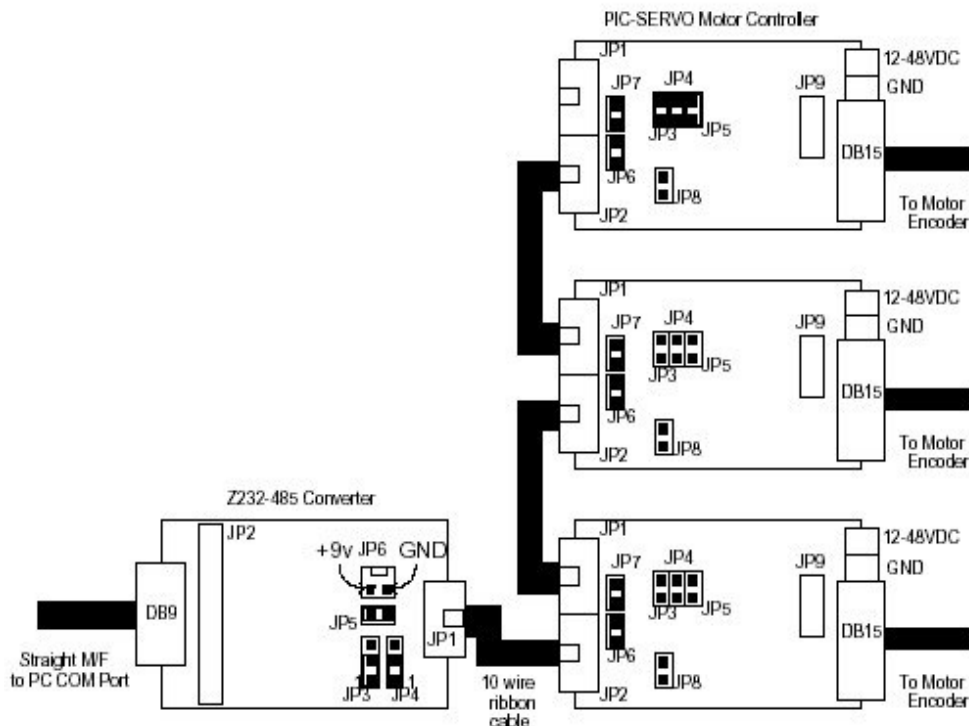
- Una fuente de 9 V en JP6 para alimentación de la lógica.
- Una fuente de 12 a 48 VCD para alimentar al motor.



**Figura 2.4.** Conexión de un solo controlador.

La conexión básica puede ser ampliada para hasta 32 controladores. Sin embargo, en nuestro caso sólo requeriremos hasta 3, como se muestra en la figura 2.5.

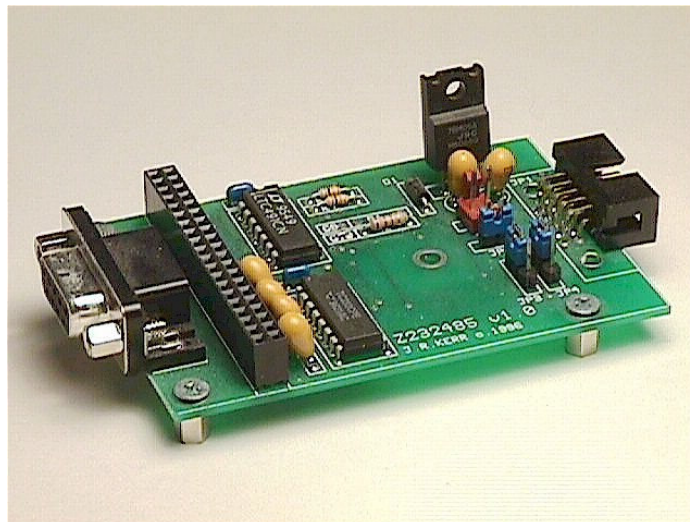
<sup>6</sup> Debe su nombre a la forma que tiene de una letra “D” y al número de patitas o conectores [4].



**Figura 2.5.** Conexión de tres controladores.

### 2.2.3. Adaptador para puerto serie

El adaptador para puerto serie utilizado para lograr la comunicación entre la computadora host y los módulos de control, es el Z232-485, el cual se muestra en la figura 2.6.



**Figura 2.6.** Adaptador para puerto serie.



Este convertidor de puerto serie<sup>7</sup> es un adaptador full-duplex<sup>8</sup> RS232-RS485, con un conector compatible al puerto COM de la computadora y al conector NMC de la red de Módulos (PIC-SERVO). Este convertidor está diseñado para soportar hasta 32 módulos NMC con índices de velocidad desde 19,200 hasta 115,200 baudios.

Para realizar la conexión con la PC, es necesario un cable con un conector hembra-macho DB9 y éste conectado con la tarjeta Z232-485. Para la conexión de la tarjeta con los módulos de control, es necesario un cable de 10 hilos plano IDC estándar, tomando en cuenta que la longitud del cable no debe exceder los 3 metros.

### 2.2.4. Palpador

En una MMC, existe una pieza de trabajo fundamental que realiza lo que se llama medición por contacto. Se le conoce como palpador y es una esfera que entra en contacto con la pieza a medir, como fue descrito en 1.1.2 [17]. La fuerza de contacto crece gradualmente hasta que se genera una señal de disparo,  $p_i(t)$  que se activa cuando la fuerza excede el umbral en el interruptor eléctrico integrado al sistema de sensado. Nuestro software utiliza esa señal para registrar las coordenadas espaciales en el instante de disparo. Cabe señalar entonces que, una vez que entran en contacto la esfera y la pieza a medir, el palpador continúa viajando en la dirección del movimiento debido a la inercia mecánica del sistema, e inclusive rebasando el instante en el cual se realizó el disparo. La distancia recorrida entre el instante de contacto y el instante de disparo constituye el error de pre-recorrido. Tal error se debe a la fuerza de doblamiento natural del palpador y de los interruptores eléctricos. Adicionalmente, el software en la PC debe proporcionar un medio de control autónomo para retirar el palpador de la pieza de trabajo en sentido opuesto al del viaje del palpador.

Cuando el sistema ha generado la señal de disparo, las coordenadas registradas corresponden al centro de la esfera del palpador,  $\mathbf{C}$ . No obstante, el punto exacto de contacto es  $\mathbf{M}$ , como se muestra en la figura 2.7. La dirección de movimiento,  $x$ -, y el instante de disparo,  $t+\delta t$ , son conocidos precisamente por el software de control en la PC que originó el movimiento y por el interruptor que accionó la señal de disparo. Si se asume que la dirección de movimiento es normal a la

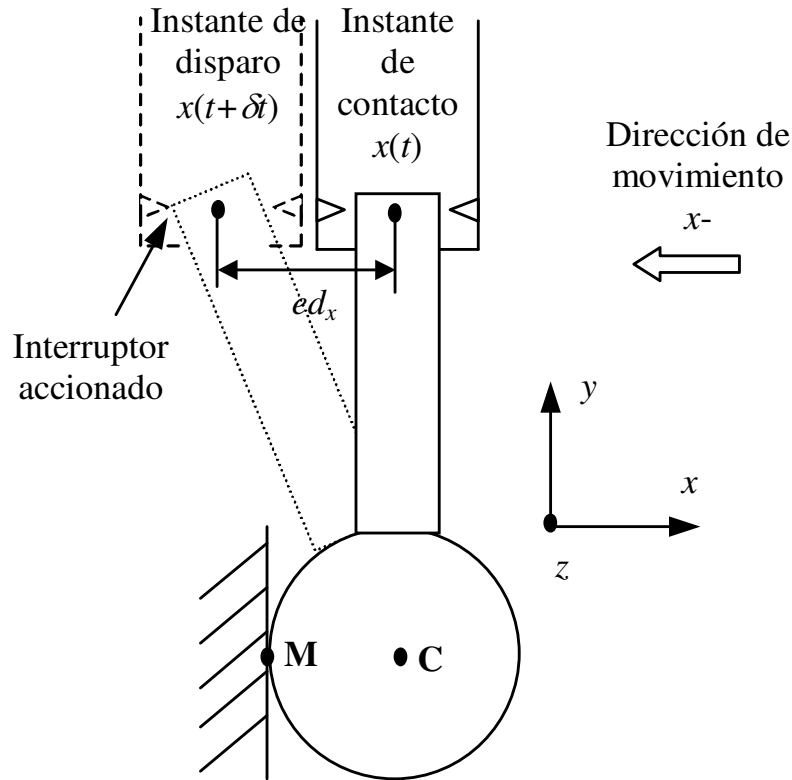
---

<sup>7</sup> Un puerto es el lugar donde la computadora se comunica con otros dispositivos. Existen los de entrada y salida, dependiendo si los datos son enviados o introducidos. El puerto serie es aquel que envía y recibe los datos Bit por Bit [7].

<sup>8</sup> Comunicación de datos que se mantiene bidireccional y simultáneamente y cuyos caracteres enviados a la computadora remota se visualizan en la pantalla de la computadora que los envía [9].

tangente de la superficie de la pieza de trabajo y que ocurre en un sólo eje coordenado, por ejemplo en  $x$ , la compensación de los errores por radio del palpador,  $r$ , y por el pre-recorrido en  $x$ ,  $ed_x$ , es la siguiente

$$M_x = x(t + \delta t) + ed_x - r \quad (2.1)$$



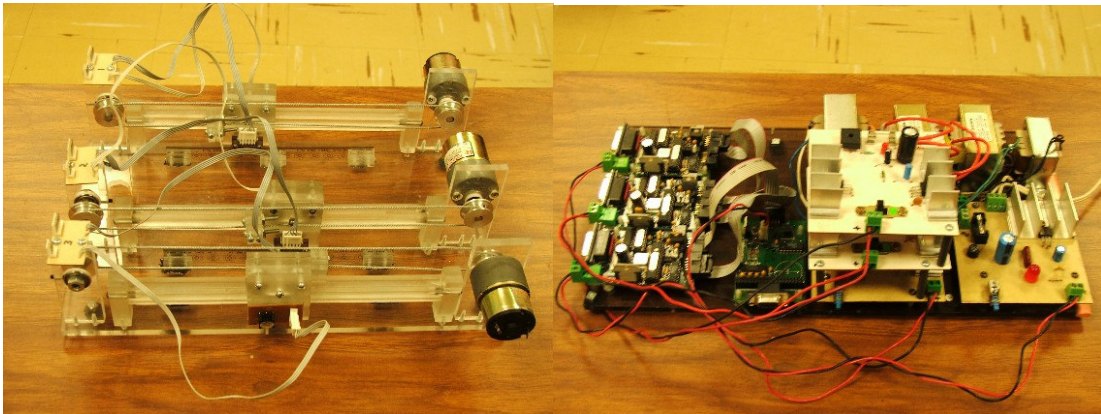
**Figura 2.7.** Medición por contacto realizada por el palpador.

De forma similar se pueden deducir las relaciones para palpaciones en los restantes ejes con sus respectivas direcciones de movimiento. El error por pre-recorrido es del tipo determinístico y se puede obtener de forma experimental.

En particular, los signos de los errores son proporcionados por la dirección de movimiento y la lectura en el instante del disparo es proporcionada por los interruptores del palpador conectados a las entradas  $p_i(t)$ .

### 2.2.5. Prototipo electro-mecánico

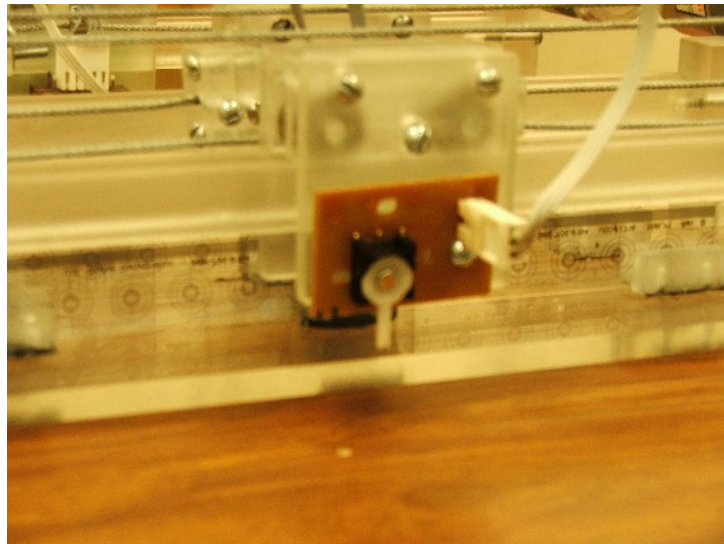
Para integrar los dispositivos hardware mencionados en 2.2.1 a 2.2.4 en un módulo que simule una MMC funcional, se construyó el prototipo de la figura 2.8.



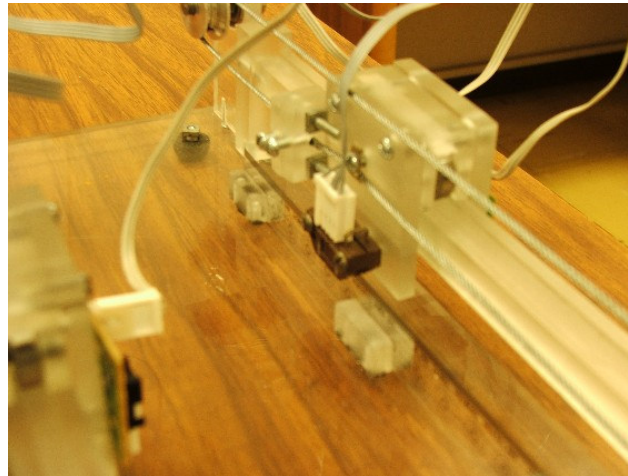
**Figura 2.8.** Prototipo electro-mecánico.

El prototipo desarrollado emula las funciones principales de una MMC como son:

- Palpador con cinco niveles de palpación, figura 2.9.
- Tres motores CD.
- Tres codificadores ópticos lineales, figura 2.10.
- Tres ejes coordenados.
- Tres controladores de servomotores.
- Tres fuentes de alimentación de potencia de 12 VCD.
- Una fuente de alimentación 9 VCD.



**Figura 2.9.** Implementación del palpador.



**Figura 2.10.** Codificador óptico lineal.

En el anexo A.1 se encuentran los diagramas mecánicos del prototipo; en el anexo A.2 los diagramas electrónicos; y en el anexo A.3 se muestran los diagramas de los conectores.

### 2.3. Descripción software para el operador

En las secciones 2.1 y 2.2 se describieron el sistema propuesto y los dispositivos que componen el prototipo de MMC así como la manera en que estos logran interactuar entre sí. El sistema se complementa con la interfase software que lleva a cabo el control de movimiento del prototipo y la interfase de usuario. En el presente apartado se describirá la forma en que se desarrolló el software que implementa lo descrito y soporta los dispositivos periféricos que complementan el sistema operando en tiempo real y con una mínima intervención por parte del operador.

El software desarrollado se realizó con Visual C++ versión 6.0 debido a que este lenguaje presentó la robustez, la confiabilidad y el entorno amigable que tanto programador como usuario final requieren [8, 13]. El entorno de desarrollo de Visual C++ 6.0 ofrece muchas posibilidades al programador, desde crear aplicaciones de diversos formatos y características hasta interactuar con dispositivos periféricos, como en éste caso es el puerto serie. Todo el IDE<sup>9</sup> en conjunto es una gran "cocina" para desarrollar las aplicaciones más versátiles tanto para entorno Windows como DOS.

Visual C++ se suministra junto con la biblioteca MFC<sup>10</sup>, que es un conjunto de clases predefinidas de Microsoft y que permiten gestionar

---

<sup>9</sup> Entorno de desarrollo

<sup>10</sup> Microsoft Foundation Classes

y crear de manera intuitiva componentes típicos de Windows. Estas clases han pasado a ser un estándar de desarrollo para aplicaciones Windows y poseen clases que facilitan la programación en C++, por ejemplo al incluir un tipo de dato CString para declarar cadenas; cosa que no existe en C y siempre es un dolor de cabeza.

También se pueden crear programas Windows que no usen MFC, así se obtendría una aplicación con la cual no haría falta distribuir la librería MFC.

Una aplicación de Windows puede ser SDI (Single Document Interface), MDI (Multiple Document Interface) o Dialog Based (basada en diálogos). Single Document es muy similar a Dialog Based, salvo que por defecto Visual C++ le agrega menús y MDI es una aplicación como por ejemplo Word, con una ventana principal que puede contener a muchas otras en su interior (como un contenedor).

Para nuestra aplicación, se eligió utilizar el SDI debido a que sólo presentamos un documento y no varios a la vez. Esto se realizó mediante el asistente MFC AppWizard<sup>11</sup>, el cual guía al programador en la creación del proyecto y nos genera un “armazón”. Es una manera práctica y muy útil de crear aplicaciones.

Algunas de las tareas principales del software desarrollado específicamente para el presente proyecto son las siguientes:

- Controlar la posición del palpador del prototipo mecánico orientado a una MMC.
- Registrar la posición de contacto del palpador con un objeto.
- Interactuar con el joystick de manera que se pueda controlar la velocidad del palpador con dicho dispositivo.

Los requerimientos mínimos para ejecutar y compilar el programa son los siguientes:

- Computadora Pentium III o superior, 64MB en RAM, 40GB HD, puerto serie RS232, Windows 2000/XP.
- Joystick analógico.
- Circuitos electrónicos para el control de movimiento.
- Fuentes de alimentación, 12V/4A, 9V/1A.
- Prototipo electro-mecánico.

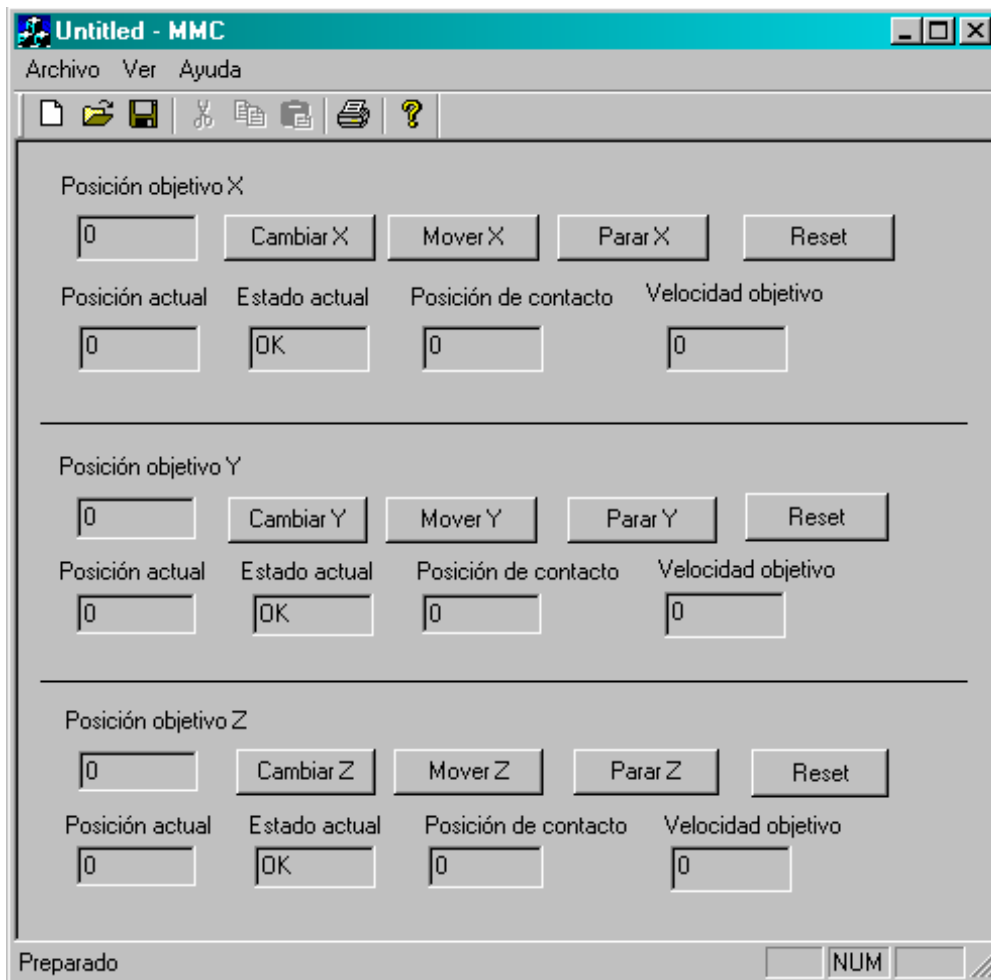
---

<sup>11</sup> Es el generador de aplicaciones propiamente dicho. Con él podemos generar esqueletos de programas para Windows basados en la MFC [14].

- Visual C++ versión 6.0

### 2.3.1. Pantalla principal

Cómo su nombre lo indica, ésta es la presentación que verá el usuario al momento de ejecutar el programa y la cual incluye todas las opciones a las que se puede tener acceso para llevar a cabo el control de la MMC. En la figura 2.11, se ejemplifica lo que el usuario verá al momento de ejecutar la aplicación llamada “MMC”.



**Figura 2.11.** Pantalla principal.

En la barra de herramientas hay tres menús: “Archivo”, “Ver”, “Ayuda”. Estos se describirán más adelante. En el cuerpo de la aplicación existen cinco cajas de texto: “Posición objetivo X”, “Posición actual”, “Estado actual”, “Posición de contacto” y “Velocidad objetivo”, y cuatro botones: “Cambiar X”, “Mover X”, “Parar X” y “Reset”, para cada eje coordenado X, Y y Z.

- **Posición objetivo.** Indica la posición a la que se desea llevar mecanismo. Ésta se puede cambiar haciendo clic en el botón “Cambiar”. Una vez cargado el dato de la nueva posición, con el botón “Mover” se ejecuta tal instrucción.
- **Posición actual.** Esta es la posición en la que se encuentra el mecanismo y cambiará en tiempo real al trasladarnos a través del eje.
- **Estado actual.** Es el encargado de verificar el funcionamiento del interruptor o palpador. El estado OK muestra que ha sido habilitado y que está en espera de ser activado. Si el interruptor hace contacto con algún objeto, se mostrará la leyenda SW1 o SW2 según sea la dirección en la que fue accionado.
- **Posición de contacto.** Caja de texto que se encarga de mostrar el dato de la posición en el momento en que el interruptor se activa debido al contacto con algún objeto.
- **Velocidad objetivo.** Despliega la velocidad a la que se está desplazando el mecanismo impulsado por el motor mediante el uso del joystick. El rango oscila entre el 0 y el 255, siendo este último la máxima velocidad.
- **Cambiar.** Botón que, al ser activado, ejecuta la ventana de diálogo de la figura 2.12 en la que es posible modificar la posición que se requiere alcanzar (posición objetivo).
- **Mover X.** Una vez modificada la posición objetivo, el programa ejecutará y llevará al motor a esa posición, haciendo clic en este botón.
- **Parar X.** Esta función ayuda a detener el movimiento del mecanismo impulsado por el motor en cualquier instante.
- **Reset.** Regresa el contador de la caja de texto “Posición actual” a cero. Es útil para establecer un origen en cualquier lugar del eje.

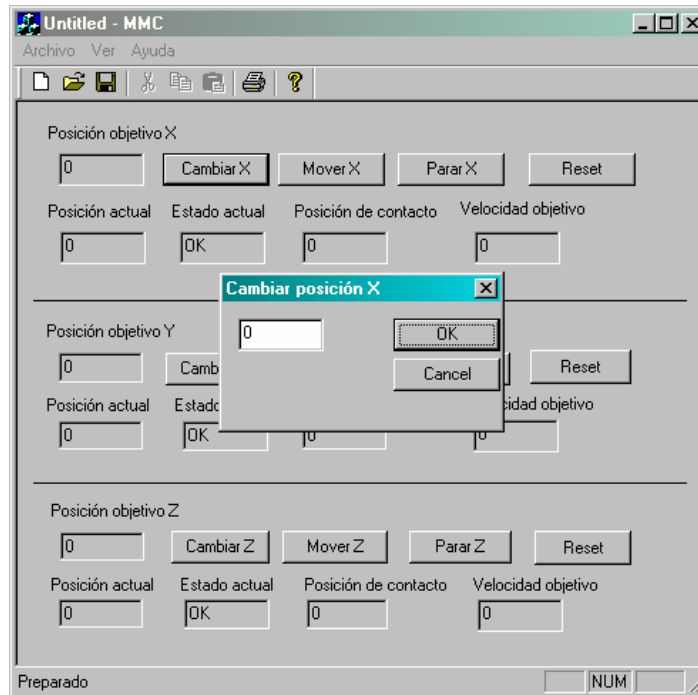


Figura 2.12. Ventana de diálogo “Cambiar posición”.

### 2.3.2. Menú Archivo

Este menú sólo consta de una función llamada *Salir* y como su nombre lo indica, sirve para terminar con la aplicación, figura 2.13.

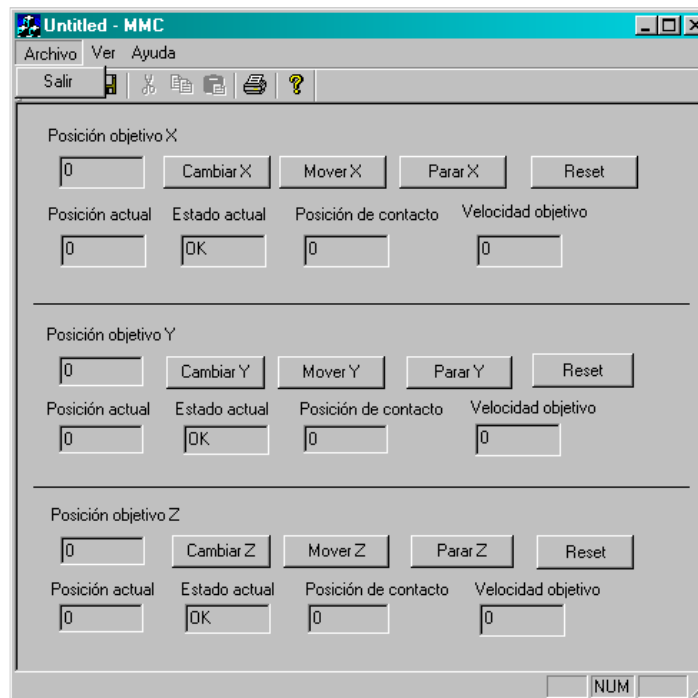


Figura 2.13. Menú “Archivo”.



### 2.3.3. Menú Ver

Dentro de este menú, se incluyen dos opciones a desplegar u ocultar: la *Barra de herramientas* y la *Barra de estado*, figura 2.14.

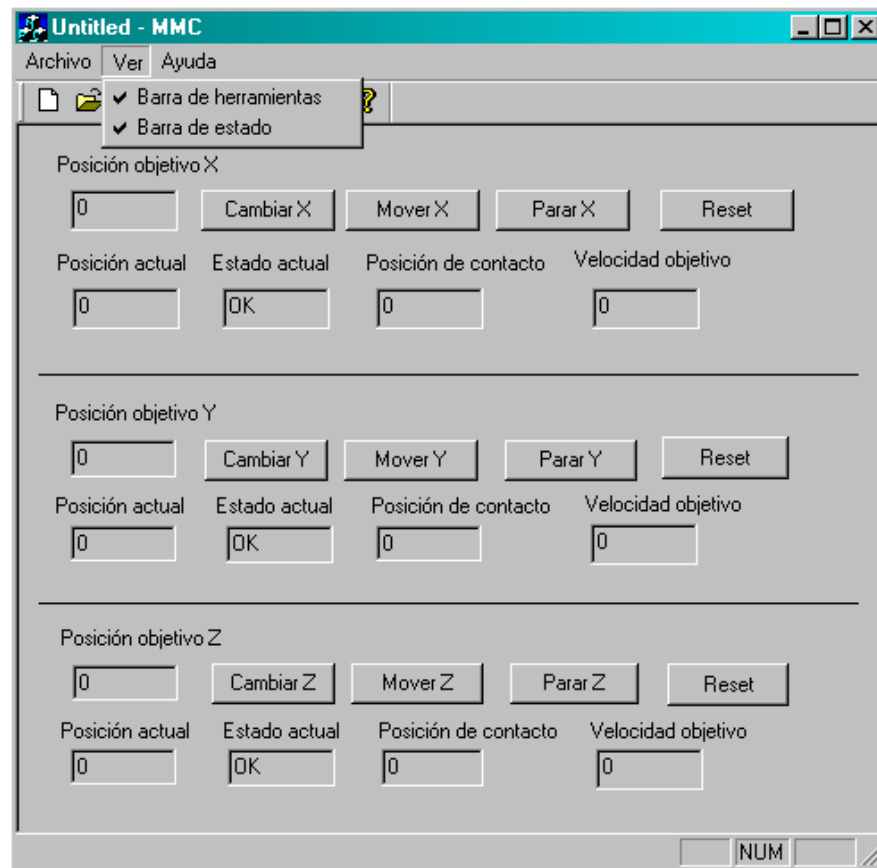
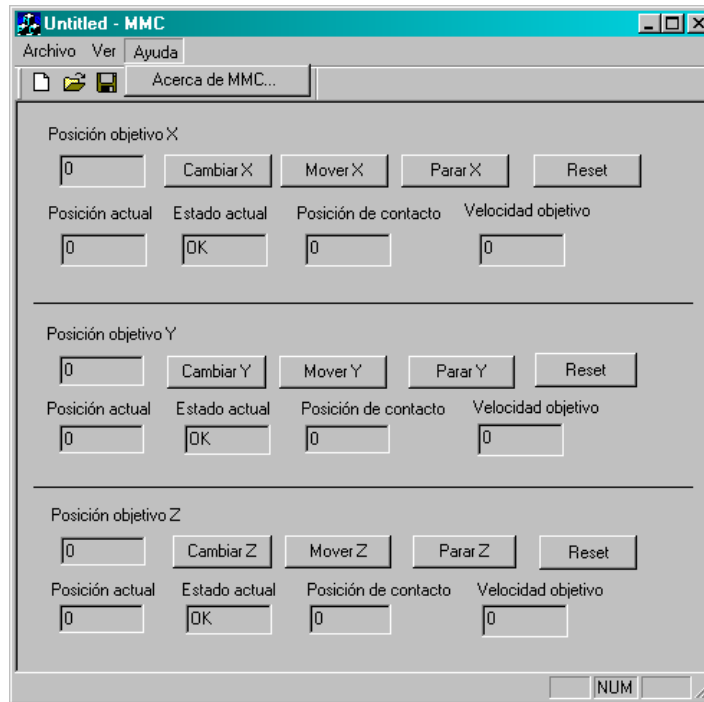


Figura 2.14. Menú “Ver”.

- **Barra de herramientas.** Es la barra que aparece debajo de los menús.
- **Barra de estado.** Nos indica si el programa está preparado o no para realizar su función. Ésta se encuentra en la parte inferior de la aplicación.

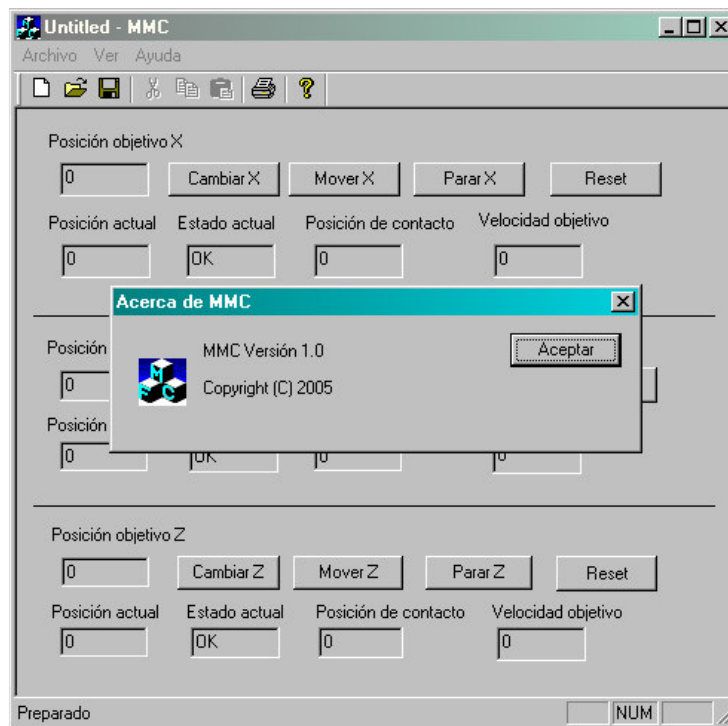
### 2.3.4. Menú Ayuda

Finalmente, encontraremos el menú “Ayuda”, el cual abrirá la ventana de diálogo “Acerca de MMC”, cuya información consiste en describir el nombre de la aplicación, la versión, así como el año en la que se desarrolló, figura 2.15.



**Figura 2.15.** Menú “Ayuda”.

En la figura 2.16 se observa la caja desplegada con toda la información del programa, una vez que se ha dado clic en la opción ya mencionada.

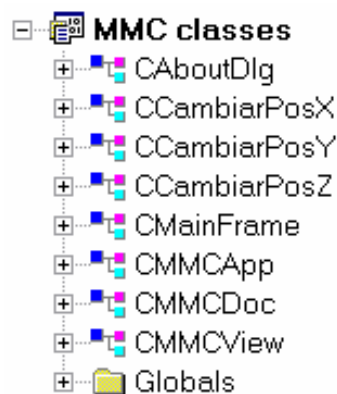


**Figura 2.16.** Ventana de diálogo que despliega el menú “Ayuda”.

## 2.4. Descripción software para el programador

La presente sección constituye una guía de la programación con el objeto de proporcionar facilidades en el mantenimiento de la aplicación “MMC”.

El programa final se organizó en ocho clases<sup>12</sup> y éstas a su vez se crearon sobre la base de la arquitectura SDI utilizando la MFC y el asistente para la creación de proyectos AppWizard de Visual C++ 6.0. La figura 2.17 muestra las clases, algunas desarrolladas específicamente para esta aplicación y otras creadas por omisión por el asistente.

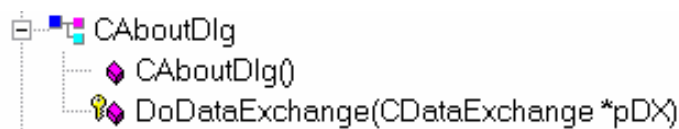


**Figura 2.17.** Clases utilizadas en el software desarrollado.

A continuación se describirán cada una de las clases utilizadas y la función que realizan. El código fuente se omite debido a que éste es de gran extensión.

### 2.4.1. Clase CAboutDlg

Esta clase, figura 2.18, es utilizada para desplegar la caja de diálogo “Acerca de”, la cual contiene información breve del programa y cuyos métodos<sup>13</sup> se describen en la tabla 2.1.



**Figura 2.18.** Clase CAboutDlg.

<sup>12</sup> Una clase es la definición de un tipo de objetos que contiene tanto métodos como variables.

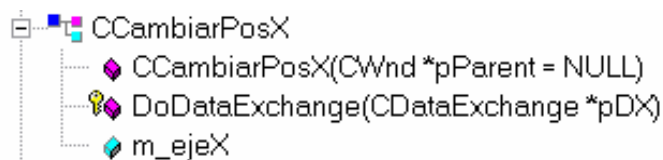
<sup>13</sup> Un método se implementa dentro de un objeto y determina cómo tiene que actuar dicho objeto cuando se produce el mensaje asociado.

Nombre	Tipo	Parámetros	Descripción
CAboutDlg	-	void	Es el constructor por omisión de la clase. No realiza inicializaciones adicionales.
DoDataExchange	void	CDataExchange *pDX	Permite el intercambio de datos con otras clases.

**Tabla 2.1.** Clase CAboutDlg.

### 2.4.2. Clase CCambiarPosX

Como su nombre lo indica, esta clase es la encargada de manejar el evento cuando se requiere cambiar o modificar la posición en el eje *x*. Aunque el asistente ayuda a crear la plantilla general del código de programación, tanto esta clase como las dos siguientes (CCambiarPosY y CCambiarPosZ), fueron añadidas y modificadas según lo requirieron. En la figura 2.19 se pueden ver los métodos que utiliza y los cuales se explican en la tabla 2.2.



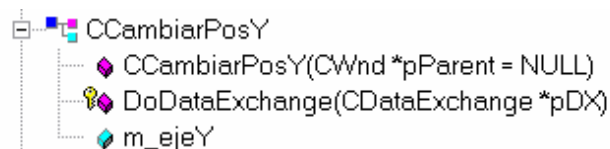
**Figura 2.19.** Clase CCambiarPosX.

Nombre	Tipo	Parámetros	Descripción
CCambiarPosX	-	CWnd *pParent=NULL	Es el constructor por omisión de la clase. No realiza inicializaciones adicionales.
DoDataExchange	void	CDataExchange *pDX	Permite el intercambio de datos con otras clases.
m_ejeX	long	-	Es la variable que utiliza esta clase.

**Tabla 2.2.** Clase CCambiarPosX.

### 2.4.3. Clase CCambiarPosY

De manera similar a la clase anterior, ésta realiza exactamente lo mismo, sólo que se utilizó para controlar el segundo eje también llamado *y*. La figura 2.20 muestra la clase y la tabla 2.3 explica sus métodos.



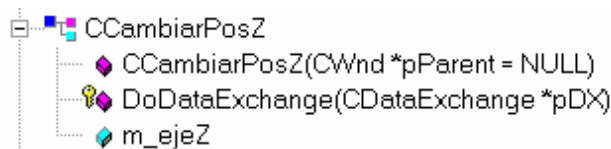
**Figura 2.20.** Clase CCambiarPosY.

Nombre	Tipo	Parámetros	Descripción
CCambiarPosY	-	CWnd *pParent=NULL	Es el constructor por omisión de la clase. No realiza inicializaciones adicionales.
DoDataExchange	void	CDataExchange *pDX	Permite el intercambio de datos con otras clases.
m_ejeY	long	-	Es la variable que utiliza esta clase.

**Tabla 2.3.** Clase CCambiarPosY.

### 2.4.4. Clase CCambiarPosZ

Finalmente, para el eje z, se creó una tercera clase que realiza exactamente las mismas funciones que las descritas con anterioridad. La figura 2.21 muestra la clase y la tabla 2.4 explica sus métodos.



**Figura 2.21.** Clase CCambiarPosZ.

Nombre	Tipo	Parámetros	Descripción
CCambiarPosZ	-	CWnd *pParent=NULL	Es el constructor por omisión de la clase. No realiza inicializaciones adicionales.
DoDataExchange	void	CDataExchange *pDX	Permite el intercambio de datos con otras clases.
m_ejeZ	long	-	Es la variable que utiliza esta clase.

**Tabla 2.4.** Clase CCambiarPosZ.

### 2.4.5. Clase CMainFrame

La clase principal es la llamada CMainFrame, puesto que ella despliega toda la aplicación en forma de caja de diálogo. Sus métodos se muestran en la figura 2.22 y se explican con más detalle en la tabla 2.5.



**Figura 2.22.** Clase CMainFrame.

Nombre	Tipo	Parámetros	Descripción
AssertValid	void	void	Valida la clase. Manda al método AssertValid() de la clase CFrameWnd.
CMainFrame	-	void	Es el constructor por omisión.
~CMainFrame	-	void	Es el destructor de la clase.
Dump	void	CDumpContext & dc	Libera el contexto de dispositivo.
OnCreate	int	LPCREATESTRUCT lpCreateStruct	Es un prototipo de manejador de mensajes al crear la estructura.
PreCreateWindow	bool	CREATESTRUCT& cs	Hace modificaciones a las propiedades de la ventana.
m_wndStatusBar	CStatusBar	-	Variable para la barra de estado.
m_wndToolBar	CToolBar	-	Variable para la barra de herramientas.

Tabla 2.5. Clase CMainFrame.

#### 2.4.6. Clase CMMCAApp

Es la encargada de dar la apariencia tridimensional a los objetos y controles que se alojan en la aplicación. La figura 2.23 muestra la clase y la tabla 2.6 su descripción.

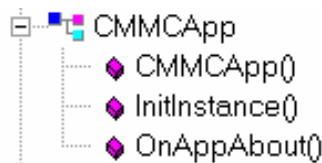


Figura 2.23. Clase CMMCAApp.

Nombre	Tipo	Parámetros	Descripción
CMMCAApp	-	void	Es el constructor por omisión de la aplicación. No realiza inicializaciones adicionales a menos que se edite la función.
InitInstance	bool	void	Realiza las siguientes inicializaciones estándar: inicia la apariencia 3D de los controles; habilita la conexión entre el documento y sus vistas.
OnAppAbout	void	void	Constituye el procedimiento para atender el evento por desplegar en la caja de diálogo "Acerca de...".

Tabla 2.6. Clase CMMCAApp.

#### 2.4.7. Clase CMMCDoc

La clase descrita en este apartado, figura 2.24, es la que tiene a su cargo la forma de almacenar los datos en archivos. Debido a que el

programa únicamente controla el movimiento de los motores, no es necesario almacenar datos adicionales. La tabla 2.7 describe los métodos creados por omisión por el asistente.



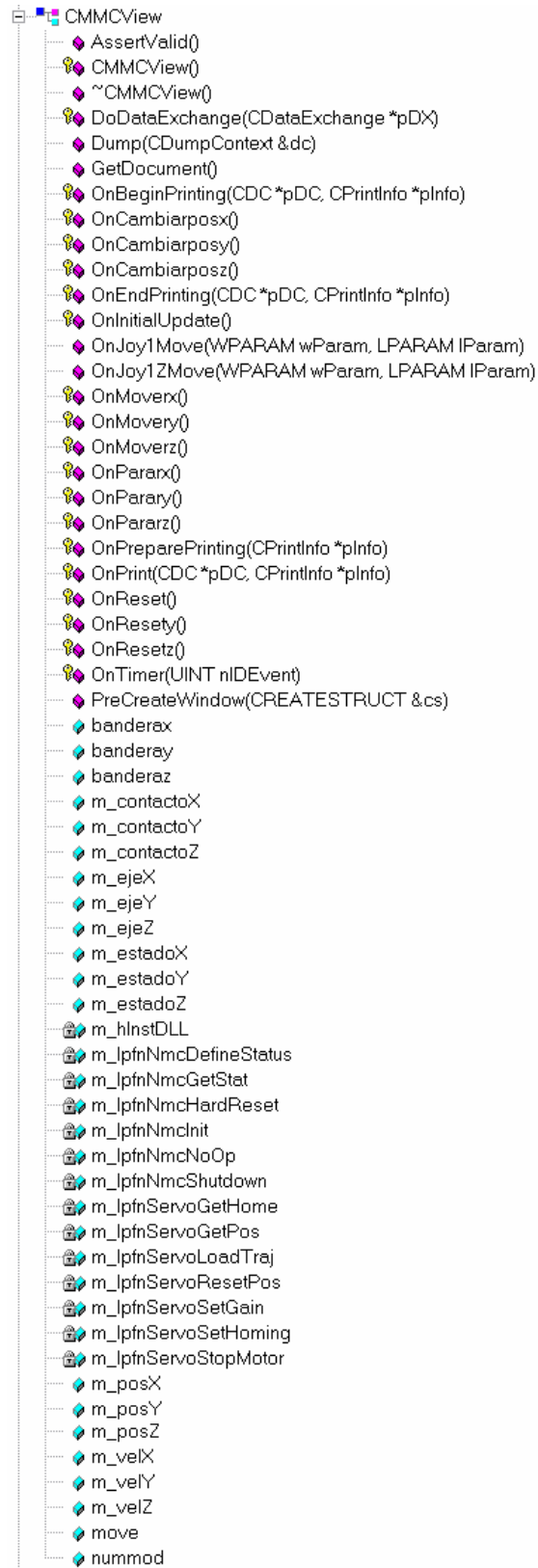
Figura 2.24. Clase CMMCDoc.

Nombre	Tipo	Parámetros	Descripción
AssertValid	void	void	Valida el documento del programa.
CMMCDoc	-	void	Es el constructor por omisión de la clase. Inicializa el formato de archivo para los datos empleados en el documento.
~CMMCDoc	-	void	Es el destructor de la clase.
Dump	void	CDumpContext &dc	Vacía la estructura de contexto del dispositivo dc.
OnNewDocument	bool	void	Inicializa los datos para crear una nueva estructura del documento para almacenar.
Serialize	void	CArchive &ar	Permite el almacenamiento de los datos de memoria con el formato de archivo empleado.

Tabla 2.7. Clase CMMCDoc.

### 2.4.8. Clase CMMCView

Esta es la clase principal del programa, ya que es donde se alojan las funciones generales del software y es en ella donde se presentan los métodos empleados para la operación del control de velocidad y posicionamiento del motor, figura 2.25. Es en esta clase en donde se hizo el mayor trabajo por parte del autor de este documento, ya que las descritas anteriormente, excepto aquellas en donde se indicó, fueron creadas por el asistente del AppWizard. La tabla 2.8 describe los métodos de la clase.



**Figura 2.25.** Clase CMMCView.



Nombre	Tipo	Parámetros	Descripción
AssertValid	void	void	Valida la clase. Llama al método CView Assert Valid().
CMMCView	-	void	Es el constructor por omisión de la clase. No realiza inicializaciones adicionales.
~CMMCView	-	void	Es el destructor de la clase.
DoDataExchange	void	CDataExchange *pDX	Aquí se aloja la construcción de los controles de la aplicación (botones, cajas de diálogo, etc.)
Dump	void	CDumpContext &dc	Libera el contexto del dispositivo dc.
GetDocument	CMMCDoc*	void	Permite el intercambio de datos que pertenecen al documento de mediciones con otras clases.
OnBeginPrinting	void	CDC *pDC CPrintInfo *pInfo	Es la instrucción para manejar el icono que aparece en la barra de herramientas y manda imprimir.
OnCambiarPosx	void	void	Despliega la caja de diálogo para cambiar la posición en el eje x.
OnCambiarPosy	void	void	Despliega la caja de diálogo para cambiar la posición en el eje y.
OnCambiarPosz	void	void	Despliega la caja de diálogo para cambiar la posición en el eje z.
OnEndPrinting	void	CDC *pDC CPrintInfo *pInfo	Permite verificar que la función OnBegingPrinting ha terminado.
OnInitialUpdate	void	void	Establece las características de una aplicación al iniciarse ésta. Abre el puerto de comunicaciones y verifica si existe algún joystick instalado y lo captura de ser así.
OnJoy1Move	LRESULT	WPARAM wParam LPARAM lParam	Permite el manejo del joystick y adecúa los movimientos para mover los motores en los ejes x y y.
OnJoy1ZMove	LRESULT	WPARAM wParam LPARAM lParam	Al igual que el anterior, pero este adecúa esos movimientos para gobernar en el eje z.
OnMoverx	void	void	Es el evento que mueve el motor en el eje x cuando se requiere llevarlo a una posición dada.
OnMovery	void	void	Es el evento que mueve el motor en el eje y cuando se requiere llevarlo a una posición dada.

OnMoverz	void	void	Es el evento que mueve el motor en el eje z cuando se requiere llevarlo a una posición dada.
OnParax	void	void	Detiene al motor en cualquier instante.
OnParary	void	void	Detiene al motor en cualquier instante.
OnPararz	void	void	Detiene al motor en cualquier instante.
OnPreparePrinting	bool	CPrintInfo *pInfo	Instrucción que se cerciora de que todo esté listo para imprimir.
OnPrint	void	CDC *pDC CPrintInfo *pInfo	Método utilizado al estar usando la opción de imprimir.
OnReset	void	void	Regresa el valor a cero, haciendo que la posición en la que se encuentra el motor, sea el nuevo origen.
OnResety	void	void	Regresa el valor a cero, haciendo que la posición en la que se encuentra el motor, sea el nuevo origen.
OnResetz	void	void	Regresa el valor a cero, haciendo que la posición en la que se encuentra el motor, sea el nuevo origen.
OnTimer	void	UINT nIDEVENT	Se establece un temporizador de 500 milisegundos para actualizar la posición actual del contador del codificador.
PreCreateWindow	bool	CREATESTRUCT &cs	Llama al método de la clase CView PreCreateWindow para poder crear la estructura de la ventana.
banderax	bool	-	Variable que sirve para discernir si se realiza una acción u otra.
banderay	bool	-	Variable que sirve para discernir si se realiza una acción u otra.
banderaz	bool	-	Variable que sirve para discernir si se realiza una acción u otra.
m_contactoX	long	-	Variable utilizada para indicar la posición de contacto.
m_contactoY	long	-	Variable utilizada para indicar la posición de contacto.
m_contactoZ	long	-	Variable utilizada para indicar la posición de contacto.
m_ejeX	long	-	Esta variable indica la posición objetivo a la que se desea llegar.
m_ejeY	long	-	Esta variable indica la posición objetivo a la que se desea llegar.
m_ejeZ	long	-	Esta variable indica la posición objetivo a la

			que se desea llegar.
m_estadoX	CString	-	Indica el momento en que se cierra el interruptor, así como en qué dirección fue accionado.
m_estadoY	CString	-	Indica el momento en que se cierra el interruptor, así como en qué dirección fue accionado.
m_estadoZ	CString	-	Indica el momento en que se cierra el interruptor, así como en qué dirección fue accionado.
m_hInstDLL	HInstance	-	Variable asociada a la librería del destructor de la clase CMMMCView.
m_lpfNmcDefineStatus	LPFNPMC-DEFINESTATUS	-	Variable que apunta a una instrucción de la librería contenida en nmclib04.dll
m_lpfNmcGetStat	LPFNPMC-GETSTAT	-	Variable que apunta a una instrucción de la librería contenida en nmclib04.dll
m_lpfNmcHardReset	LPFNPMC-HARDRESET	-	Variable que apunta a una instrucción de la librería contenida en nmclib04.dll
m_lpfNmcInit	LPFNPMC-INIT	-	Variable que apunta a una instrucción de la librería contenida en nmclib04.dll
m_lpfNmcNoOp	LPFNPMC-NOOP	-	Variable que apunta a una instrucción de la librería contenida en nmclib04.dll
m_lpfNmcShutdown	LPFNPMC-SHUTDOWN	-	Variable que apunta a una instrucción de la librería contenida en nmclib04.dll
m_lpfNmcServoGetHome	LPFNPMC-SERVOGETHOME	-	Variable que apunta a una instrucción de la librería contenida en nmclib04.dll
m_lpfNmcServoGetPos	LPFNPMC-SERVOGETPOS	-	Variable que apunta a una instrucción de la librería contenida en nmclib04.dll
m_lpfNmcServoLoadTraj	LPFNPMC-SERVOLOADTRAJ	-	Variable que apunta a una instrucción de la librería contenida en nmclib04.dll
m_lpfNmcServoResetPos	LPFNPMC-SERVORESETPOS	-	Variable que apunta a una instrucción de la librería contenida en nmclib04.dll
m_lpfNmcServoSetGain	LPFNPMC-SERVOSETGAIN	-	Variable que apunta a una instrucción de la librería contenida en nmclib04.dll
m_lpfNmcServoSetHoming	LPFNPMC-SERVOSETHOMING	-	Variable que apunta a una instrucción de la librería contenida en nmclib04.dll
m_lpfNmcServoStopMotor	LPFNPMC-SERVOSTOPMOTOR	-	Variable que apunta a una instrucción de la

			librería contenida en nmclib04.dll
m_posX	long	-	La posición actual es actualizada y manejada por esta variable.
m_posY	long	-	La posición actual es actualizada y manejada por esta variable.
m_posZ	long	-	La posición actual es actualizada y manejada por esta variable.
m_velX	long	-	Esta variable lleva la velocidad con la que se mueve el motor.
m_velY	long	-	Esta variable lleva la velocidad con la que se mueve el motor.
m_velZ	long	-	Esta variable lleva la velocidad con la que se mueve el motor.
move	bool	-	Actúa como bandera para comprobar si se debe ejecutar un movimiento o no.
nummod	int	-	Es la variable que lleva el conteo de los módulos conectados a la red.

**Tabla 2.8.** Clase CMMCView.

## **CAPÍTULO 3**

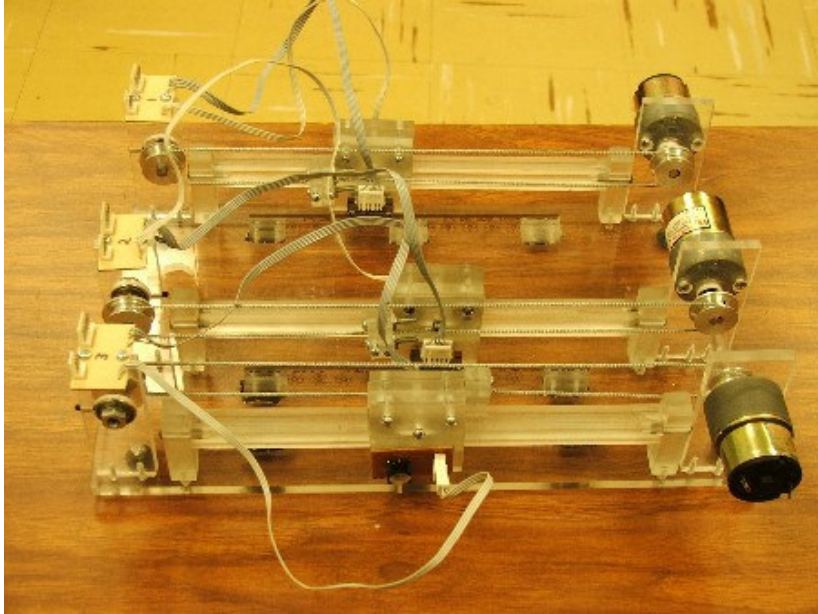
# **RESULTADOS Y CONCLUSIONES**

El presente capítulo contiene los resultados y conclusiones del presente proyecto.

### **3.1. Resultados**

Los resultados los mediremos desde dos perspectivas: una cualitativa y otra cuantitativa. La primera la ligaremos al sistema en general, es decir, a la manera en que el prototipo ha sido terminado y puesto en funcionamiento. Para la segunda perspectiva, se llevaron a cabo dos experimentos, con los que se pretendió medir la repetibilidad y la exactitud con la que puede responder el sistema en general.

En la figura 3.1, se puede apreciar el prototipo electro-mecánico, elaborado en acrílico, que emula tres ejes coordenados con palpador mecánico de cinco niveles de palpación y codificador óptico lineal de 360X4 CPI (cuentas por pulgada), resultando en una resolución de 17 $\mu$ m. En la figura 3.2 se muestra el sistema en su totalidad, incluyendo la PC y el joystick, entre otras cosas.



**Figura 3.1.** Prototipo electro-mecánico.



**Figura 3.2.** Prototipo completo de MMC.

En cuanto a los experimentos, como ya se mencionó anteriormente, se llevaron a cabo dos, con los cuales se midió la capacidad del prototipo para responder a los objetivos planteados inicialmente.

### **3.1.1. Repetibilidad en el posicionamiento**

El primero de los experimentos consistió en mandar una posición determinada mediante el software desarrollado y que ésta fuera alcanzada con la mayor precisión posible. Esta acción fue repetida

varias veces para cada uno de los ejes y en ambos sentidos. Las tablas 3.1 a 3.3 muestran cuatro columnas: 1) número de evento; 2) la posición actual; 3) la posición objetivo; 4) posición alcanzada; 5) diferencia entre la posición objetivo y la alcanzada. En las figuras 3.3, 3.5 y 3.7 se grafica la diferencia existente entre la posición actual y objetivo en cuentas por pulgada contra el número de evento. Las figuras 3.4, 3.6 y 3.8 contendrá la misma información que las gráficas anteriores, excepto que en éstas, los resultados se dan en micrómetros. Para entender ambas gráficas, entre menor sea la diferencia, mejor será la repetibilidad, esto es, entre más lecturas o eventos cercanos al cero, un mejor resultado para nuestro prototipo.

Las posiciones objetivo se eligieron al azar y se mandaron cuando el mecanismo se encontraba en diversas posiciones para así asegurar la validez del experimento.

## Eje x

Evento	Posición Actual	Posición Objetivo	Posición Alcanzada	Diferencia (cpi)	Diferencia ( $\mu\text{m}$ )
1	0	1000	1001	-1	-17
2	1001	1510	1505	5	85
3	2310	0	7	-7	-119
4	7	-100	-93	-7	-119
5	-93	-500	-494	-6	-102
6	-494	-450	-449	-1	-17
7	722	-450	-443	-7	-119
8	2448	2000	2004	-4	-68
9	-2722	0	-4	4	68
10	835	0	0	0	0
11	-2620	-2500	-2503	3	51
12	949	1000	997	3	51
13	997	-55	-51	-4	-68
14	384	554	556	-2	-34
15	556	-2	-1	-1	-17
16	-1648	501	497	4	68
17	497	-10	-6	-4	-68
18	-1534	-1332	-1339	7	119
19	3095	0	3	-3	-51
20	3	-120	-113	-7	-119

Tabla 3.1. Repetibilidad en el posicionamiento del eje x.

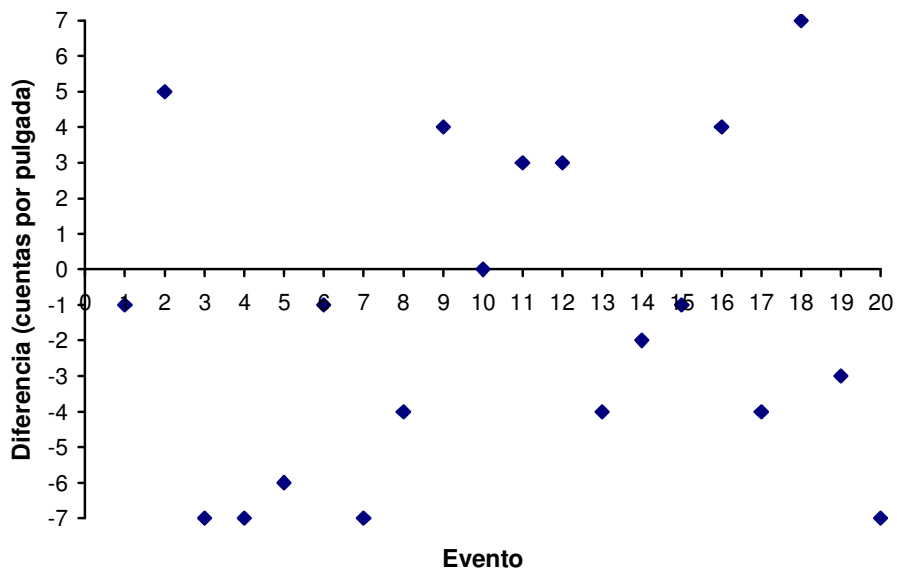
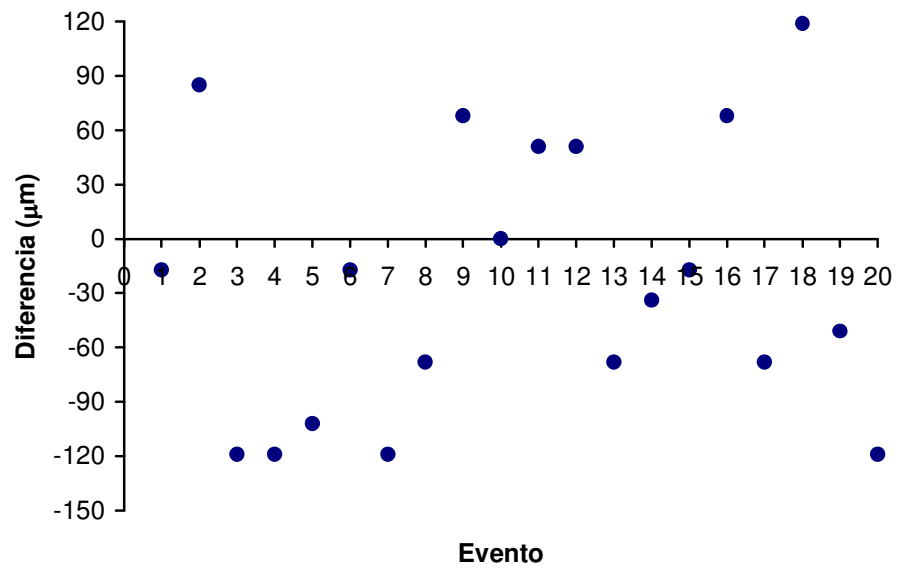


Figura 3.3. Gráfica de la repetibilidad en el eje x en cpi.





**Figura 3.4.** Gráfica de la repetibilidad en el eje  $x$  en micrómetros.

## Eje y

Evento	Posición Actual	Posición Objetivo	Posición Alcanzada	Diferencia (cpi)	Diferencia ( $\mu\text{m}$ )
1	-946	0	3	-3	-51
2	3	-650	-651	1	17
3	-651	-2000	-2002	2	34
4	840	842	841	1	17
5	1898	1000	998	2	34
6	-175	-100	-97	-3	-51
7	-97	0	2	-2	-34
8	-2135	-2000	-1998	-2	-34
9	16	2000	2001	-1	-17
10	2001	2005	2003	2	34
11	3217	-1000	-1002	2	34
12	-1002	0	2	-2	-34
13	1713	-1713	-1715	2	34
14	-1715	2135	2137	-2	-34
15	4498	4000	4003	-3	-51
16	-2259	4496	4498	-2	-34
17	2314	-100	-101	1	17
18	-2009	-2000	-2000	0	0
19	-2000	0	2	-2	-34
20	2553	2	1	1	17

Tabla 3.2. Repetibilidad en el posicionamiento del eje y.

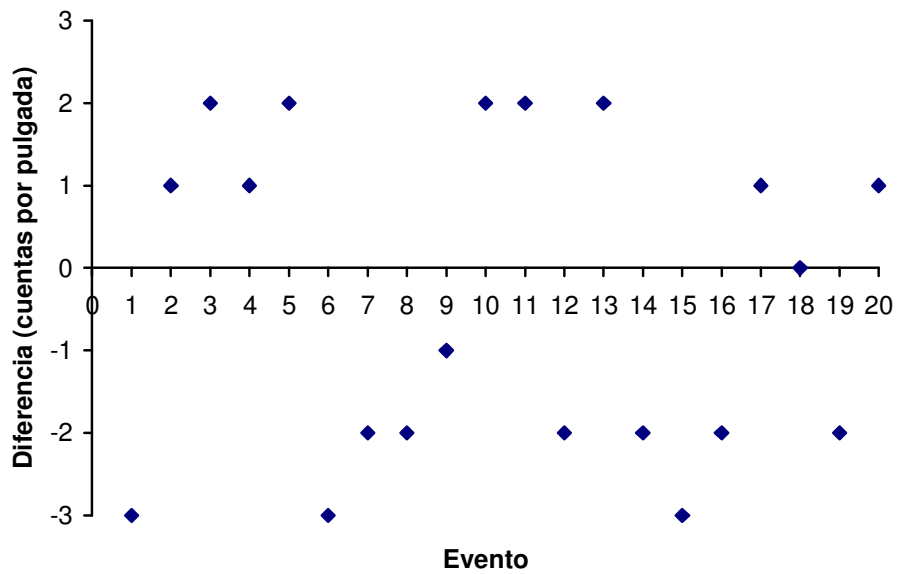
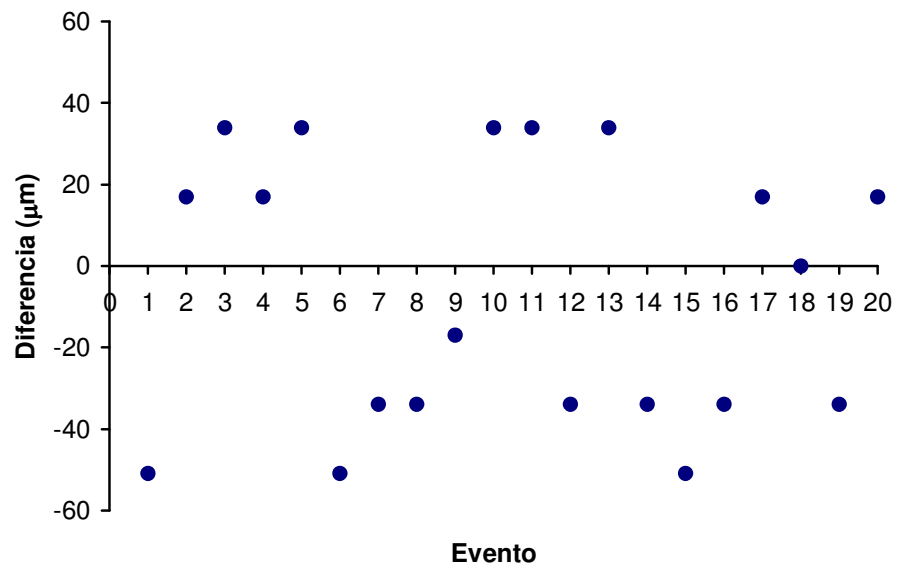


Figura 3.5. Gráfica de la repetibilidad en el eje y en cpi.



**Figura 3.6.** Gráfica de la repetibilidad en el eje y en micrómetros.

## Eje z

Evento	Posición Actual	Posición Objetivo	Posición Alcanzada	Diferencia (cpi)	Diferencia ( $\mu\text{m}$ )
1	0	236	239	-3	-51
2	239	1000	997	3	51
3	989	980	978	2	34
4	1544	1500	1505	-5	-85
5	-100	-106	-110	4	68
6	168	-200	-205	5	85
7	390	0	-4	4	68
8	-1312	-200	-205	5	85
9	-205	0	-2	2	34
10	315	300	305	-5	-85
11	1139	895	892	3	51
12	892	-100	-105	5	85
13	-650	-699	-704	5	85
14	-1634	-250	-245	-5	-85
15	-245	90	93	-3	-51
16	93	0	-4	4	68
17	-372	23	26	-3	-51
18	439	500	501	-1	-17
19	501	-2	-6	4	68
20	941	1100	1098	2	34

Tabla 3.3. Repetibilidad en el posicionamiento del eje z.

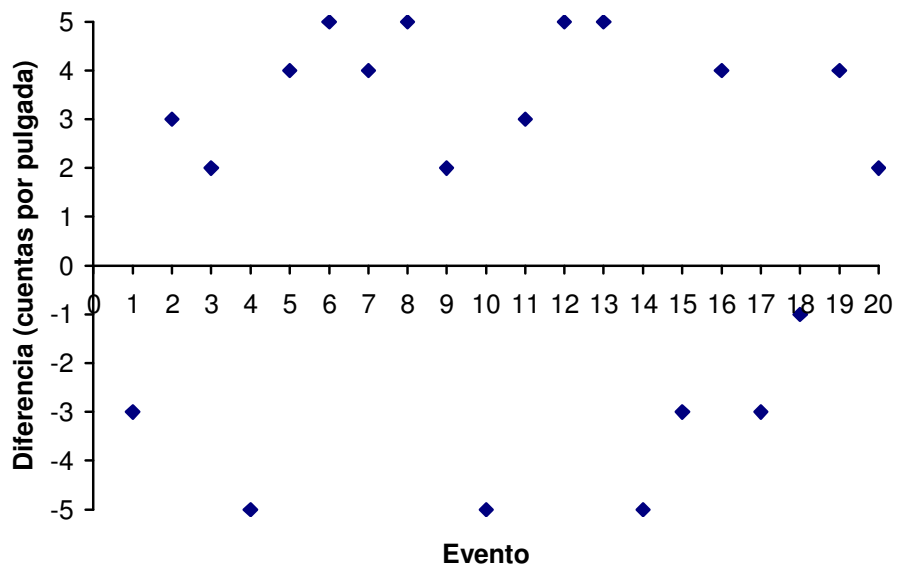
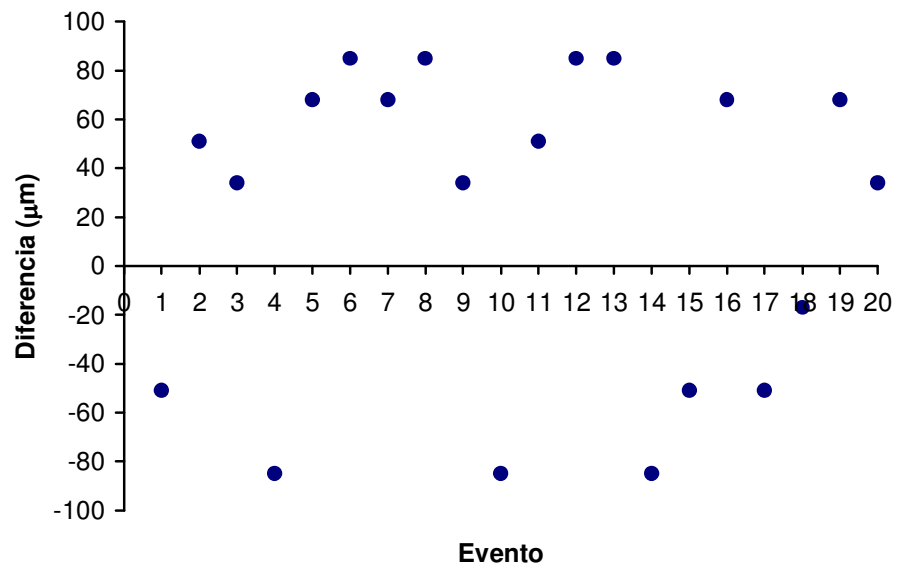


Figura 3.7. Gráfica de la repetibilidad en el eje z en cpi.



**Figura 3.8.** Gráfica de la repetibilidad en el eje  $z$  en micrómetros.

### **3.1.2. Repetibilidad en el punto de contacto**

El segundo experimento consistió en colocar un objeto en una posición determinada, llevar el interruptor (que en nuestro caso emula al palpador de una MMC) hacia el objeto con la intención de que dicho interruptor se accionara, logrando así detener el movimiento y registrar las posiciones final y de contacto. Este mismo experimento se llevó a cabo para cada uno de los ejes y en ambos sentidos de los mismos. Las tablas 3.4 a 3.9 describen lo siguiente: 1) el número de evento; 2) la posición en la que termina el mecanismo impulsado por el motor; 3) posición de contacto; 4) diferencia entre la posición de contacto y la posición en la que termina el mecanismo impulsado por el motor. Se presentan dos tablas para el caso de cada eje debido a los sentidos (positivo y negativo) de los mismos. Cabe señalar que en este caso no se puede determinar la posición exacta en la que se encontraba el objeto que entró en contacto con el interruptor. De esta manera, una vez recabadas las lecturas, se obtuvo un promedio que es el que se toma como posición patrón. Las figuras 3.9 a 3.20 muestran gráficamente el desempeño obtenido.

## Eje x (+)

Evento	Posición actual	Posición de Contacto	Diferencia (cpi)	Diferencia ( $\mu\text{m}$ )
1	3946	3921	0	0
2	3951	3918	-3	-51
3	3940	3921	0	0
4	3945	3925	4	68
5	3952	3922	1	17
6	3931	3918	-3	-51
7	3945	3920	-1	-17
8	3950	3922	1	17
9	3938	3918	-3	-51
10	3947	3921	0	0
<b>Promedio = 3921</b>				

Tabla 3.4. Repetibilidad en el eje x(+).

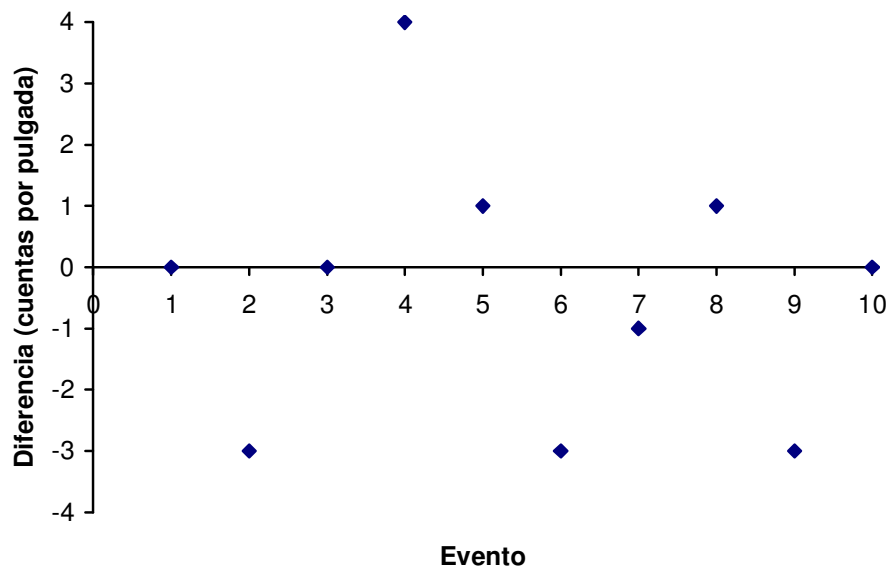
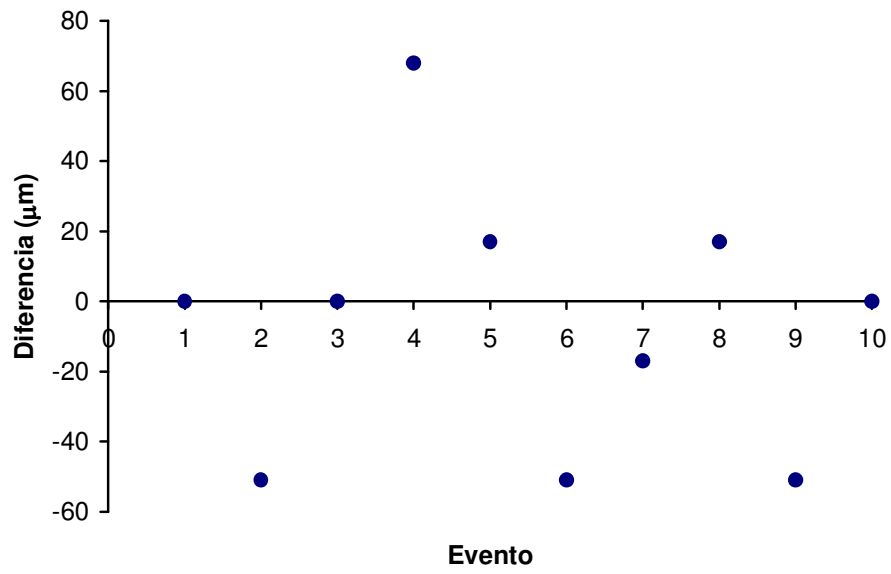


Figura 3.9. Gráfica de la repetibilidad en el eje x(+) en cpi.



**Figura 3.10.** Gráfica de la repetibilidad en el eje  $x(+)$  en micrómetros.



## Eje x (-)

Evento	Posición actual	Posición de Contacto	Diferencia (cpi)	Diferencia ( $\mu\text{m}$ )
1	-1402	-1373	-1	-17
2	-1395	-1370	2	34
3	-1398	-1375	-3	-51
4	-1401	-1372	0	0
5	-1394	-1370	2	34
6	-1394	-1371	1	17
7	-1390	-1371	1	17
8	-1393	-1370	2	34
9	-1400	-1374	-2	-34
10	-1398	-1372	0	0
<b>Promedio = -1372</b>				

Tabla 3.5. Repetibilidad en el eje x(-).

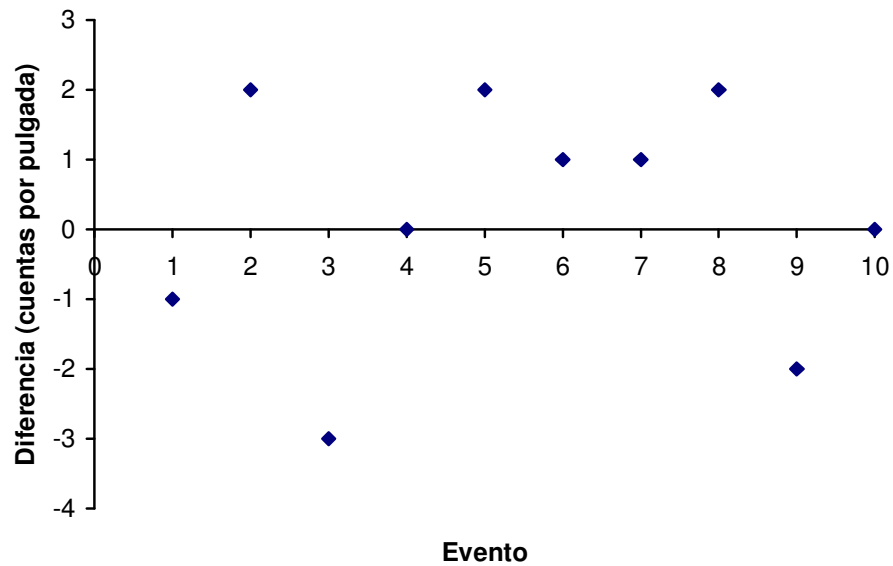
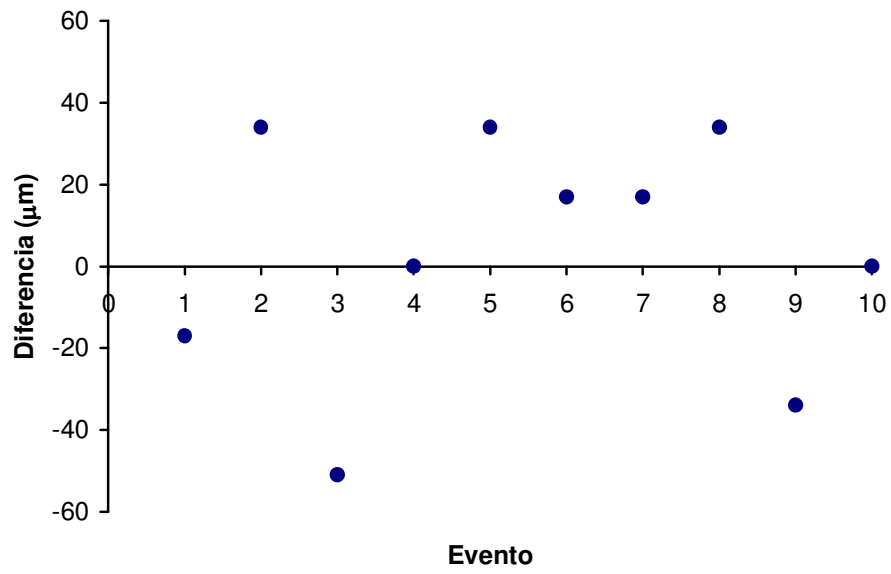


Figura 3.11. Gráfica de la repetibilidad en el eje x(-) en cpi.



**Figura 3.12.** Gráfica de la repetibilidad en el eje  $x(-)$  en micrómetros.

## Eje y (+)

Evento	Posición actual	Posición de Contacto	Diferencia (cpi)	Diferencia ( $\mu\text{m}$ )
1	2917	2570	0	0
2	2957	2572	2	34
3	2938	2566	-4	-68
4	2924	2570	0	0
5	2934	2571	1	17
6	2935	2572	2	34
7	2920	2570	0	0
8	2939	2568	-2	-34
9	2938	2570	0	0
10	2940	2572	2	34
<b>Promedio = 2570</b>				

Tabla 3.6. Repetibilidad en el eje y(+).

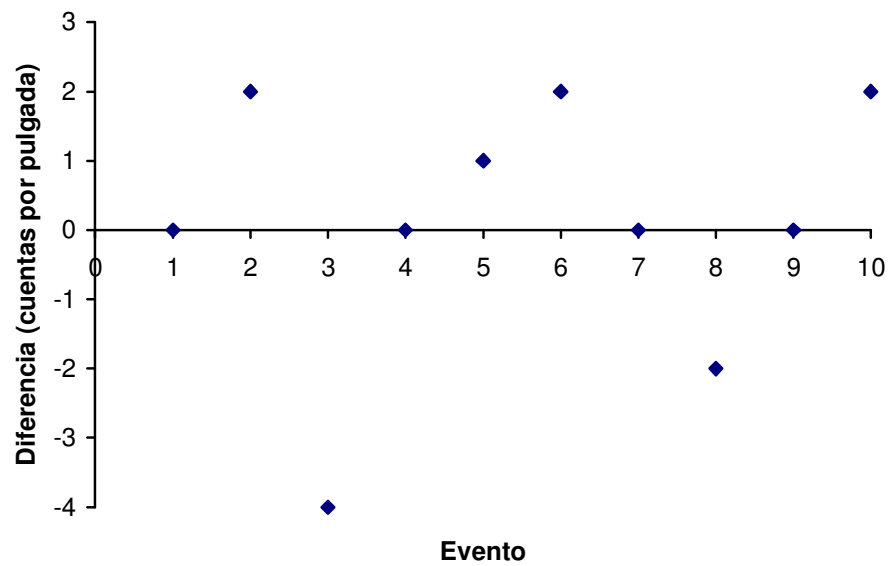
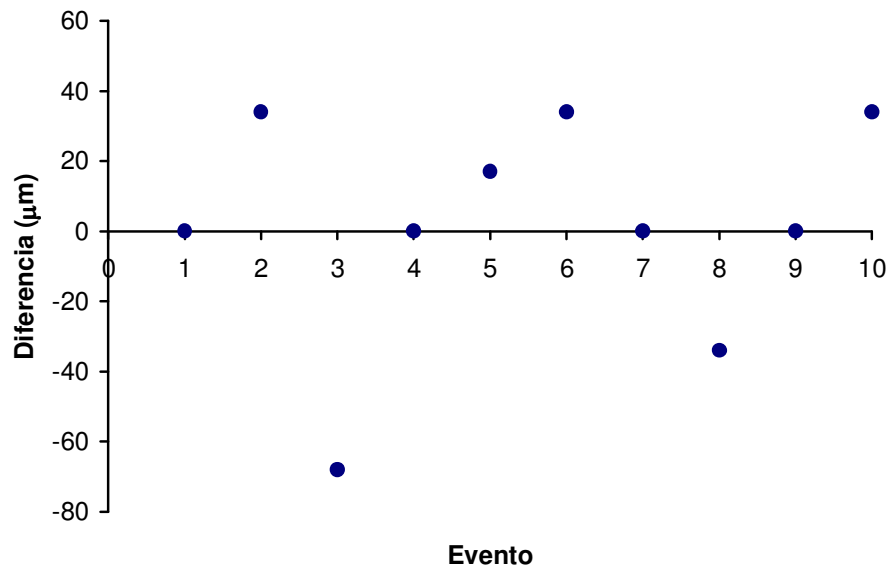


Figura 3.13. Gráfica de la repetibilidad en el eje y(+) en cpi.



**Figura 3.14.** Gráfica de la repetibilidad en el eje y(+) en micrómetros.

## Eje y (-)

Evento	Posición actual	Posición de Contacto	Diferencia (cpi)	Diferencia ( $\mu\text{m}$ )
1	-4038	-3645	2	34
2	-4046	-3648	-1	-17
3	-4040	-3647	0	0
4	-4057	-3648	-1	-17
5	-4038	-3648	-1	-17
6	-4050	-3650	-3	-51
7	-4052	-3646	1	17
8	-4038	-3647	0	0
9	-4051	-3646	1	17
10	-4050	-3648	-1	-17
<b>Promedio = -3647</b>				

Tabla 3.7. Repetibilidad en el eje y(-).

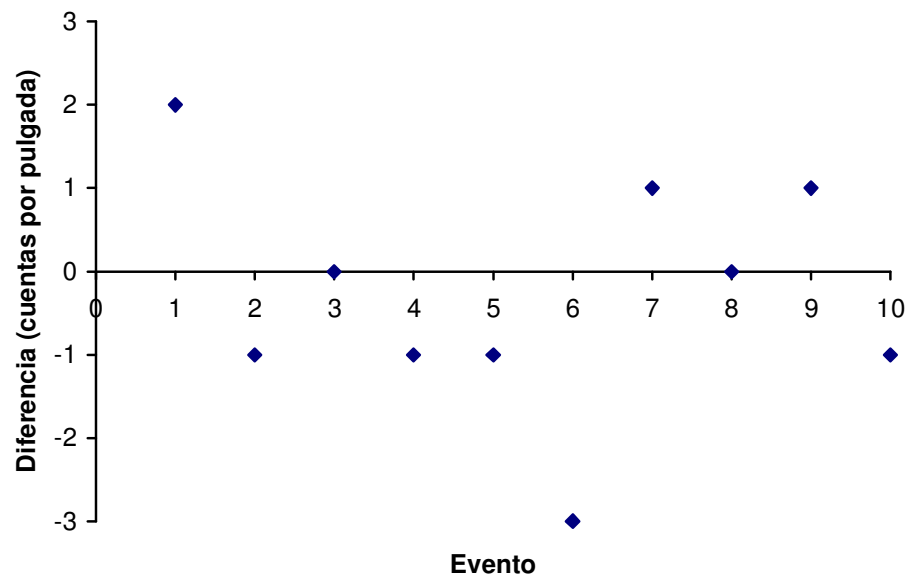
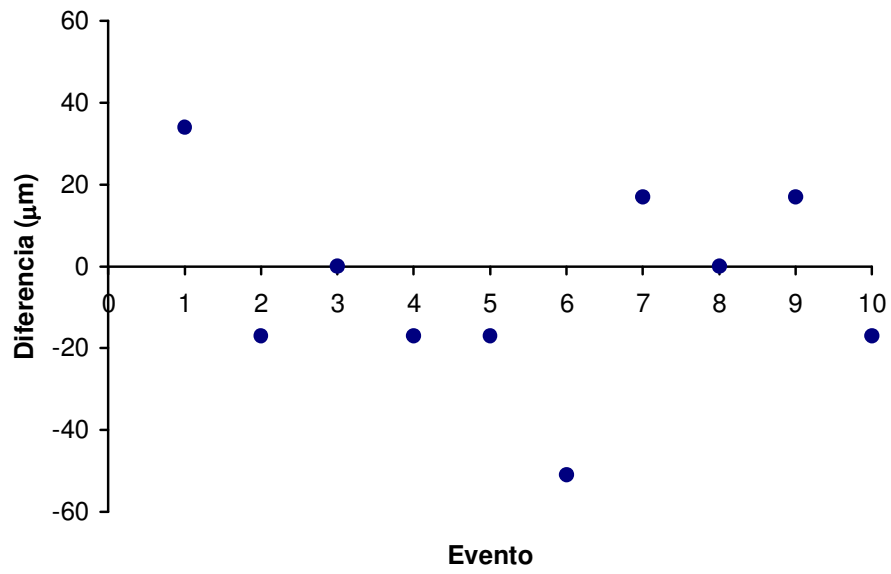


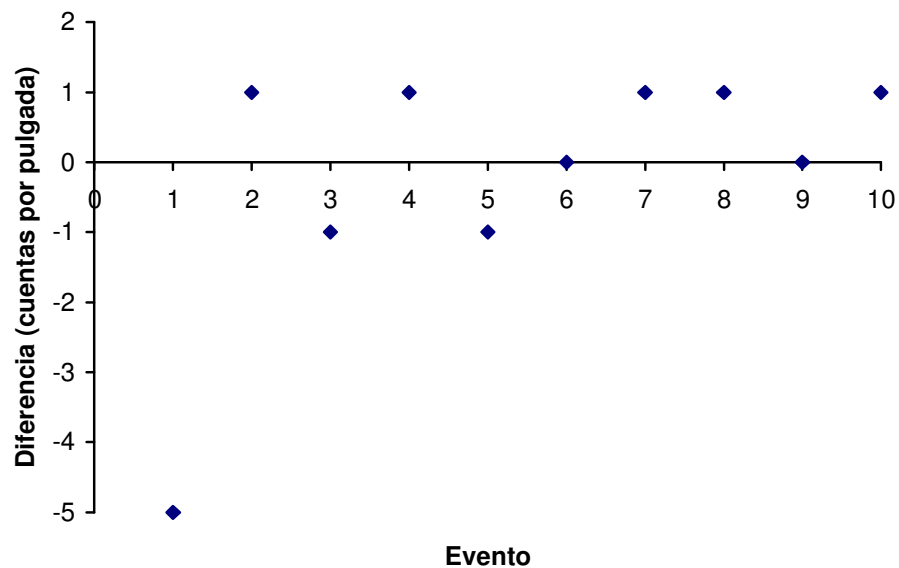
Figura 3.15. Gráfica de la repetibilidad en el eje y(-) en cpi.

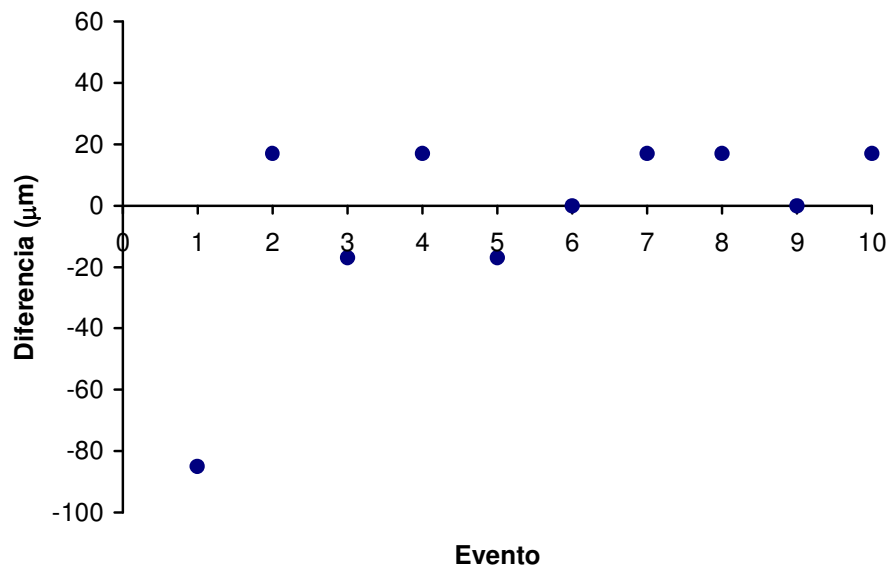


**Figura 3.16.** Gráfica de la repetibilidad en el eje y(-) en micrómetros.

**Eje z (+)**

Evento	Posición actual	Posición de Contacto	Diferencia (cpi)	Diferencia ( $\mu\text{m}$ )
1	2347	2313	-5	-85
2	2397	2319	1	17
3	2334	2317	-1	-17
4	2385	2319	1	17
5	2386	2317	-1	-17
6	2416	2318	0	0
7	2422	2319	1	17
8	2410	2319	1	17
9	2405	2318	0	0
10	2385	2319	1	17
<b>Promedio = 2318</b>				

**Tabla 3.8.** Repetibilidad en el eje z(+).**Figura 3.17.** Gráfica de la repetibilidad en el eje z(+) en cpi.



**Figura 3.18.** Gráfica de la repetibilidad en el eje  $z(+)$  en micrómetros.



## Eje z (-)

Evento	Posición actual	Posición de Contacto	Diferencia (cpi)	Diferencia ( $\mu\text{m}$ )
1	-2918	-2889	-1	-17
2	-2923	-2887	1	17
3	-2899	-2890	-2	-34
4	-2983	-2889	-1	-17
5	-2943	-2888	0	0
6	-2924	-2888	0	0
7	-2905	-2889	-1	-17
8	-2933	-2887	1	17
9	-2936	-2887	1	17
10	-2912	-2887	1	17
<b>Promedio = -2888</b>				

Tabla 3.9. Repetibilidad en el eje z(-).

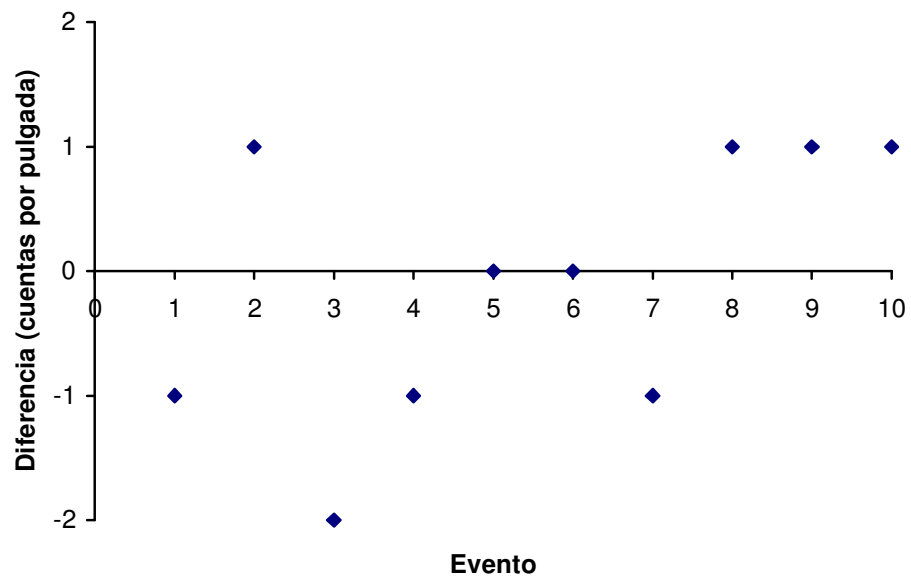
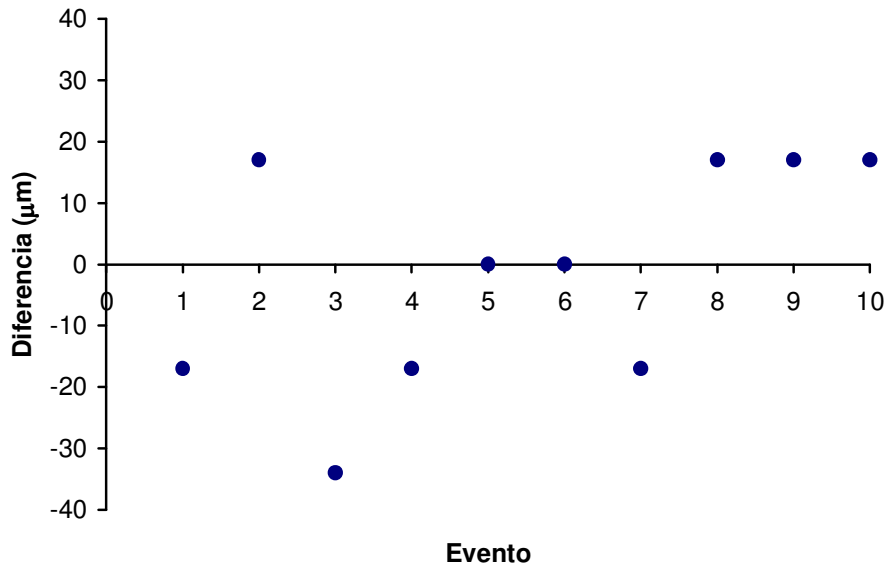


Figura 3.19. Gráfica de la repetibilidad en el eje z(-) en cpi.



**Figura 3.20.** Gráfica de la repetibilidad en el eje z(-) en micrómetros.

## 3.2. Conclusiones

Se desarrolló y se puso en funcionamiento un sistema de control para tres servomotores con interfase para PC orientado a una MMC. Dicho sistema ha sido evaluado utilizando un prototipo de máquina de tres ejes coordenados que incluye un palpador de contacto de cinco niveles de palpación y codificadores ópticos lineales. Además, se utilizaron tarjetas electrónicas comerciales para el control de lazo cerrado de servomotores que ejecutan algoritmos de control del tipo PID.

Es importante hacer notar que existieron varios factores en contra al momento de armar todo el dispositivo. Un problema bastante fuerte fueron las piezas mecánicas, tales como poleas, rieles y hasta el cable, los cuales obviamente no tienen los acabados que se necesitan para una MMC de precisión. No obstante estas complicaciones, el sistema en general respondió de una forma bastante satisfactoria.

Se desarrolló el software de operación utilizando herramientas básicas de programación como Visual C++ 6.0. El software implementa el algoritmo de medición y controla las tarjetas electrónicas en la modalidad de “maestro-esclavo”. Bajo este esquema, la PC constituye el control maestro en donde son enviados comandos básicos de control (realizar movimiento, parar motor, leer codificador, leer entradas, actualizar salidas) mediante el puerto serie RS232. Por otra parte, las

tarjetas electrónicas ejecutan el comando maestro directamente en los dispositivos actuadores o transductores. El prototipo incluye un joystick analógico con la finalidad de generar los comandos de control de movimiento de una manera amigable con el usuario.

Se llevaron a cabo dos experimentos que verifican la repetibilidad del sistema en dos enfoques sustanciales: posicionamiento y medición. Tales experimentos los podemos resumir en la tabla 3.10, tomando en cuenta que son los valores más alejados que se presentaron.

	<b>Posicionamiento</b>	<b>Medición por contacto</b>
<b>Repetibilidad (<math>\mu\text{m}</math>)</b>	$\pm 119$	$\pm 85$

**Tabla 3.10.** Repetibilidad del sistema.

Los valores de la tabla 3.10 representan el desempeño del sistema en un prototipo mecánico con problemas fuertes de diseño y dispositivos que no son adecuados para mediciones de gran exactitud. El aliciente es que estos resultados pueden ser mejorados notablemente en un diseño mecánico de precisión, como el que posee una MMC.

El prototipo de MMC, elaborado en acrílico, fue propuesto como objeto de prueba y solamente es utilizado en la depuración del software de operación y configuración de las tarjetas electrónicas de control. Por lo tanto, las limitaciones son obvias y resultan ser del tipo mecánico. Por otra parte, el sistema de control, incluyendo software y tarjetas electrónicas, demuestran su viabilidad para ser incluidos en un instrumento de medición que alcance los niveles de exactitud deseados en una MMC.

Así pues, podemos resumir que los resultados no han sido los mejores, ya que esperábamos que la exactitud fuera de  $\pm 2$  cuentas por pulgada, o bien, de  $\pm 54$  micrómetros. En ambos experimentos se excedió el límite objetivo, pero eso es debido a las diversas complicaciones y limitaciones mecánicas a las que nos debimos enfrentar y las cuales se han descrito en su momento. No obstante, este error podrá estar dentro del rango, una vez que se logre realizar un mejor prototipo mecánico.

Aún faltan algunas cosas por mejorar, así como depurar ciertos detalles. Pero por ahora, podemos quedar satisfechos de haber logrado y desarrollado un buen prototipo de MMC y demostrado que se puede desarrollar ésta tecnología a pesar de las limitaciones que ya conocemos, además de las económicas y sobre todo, sin tener que copiar la de otros países.

### **3.3. Trabajo a futuro**

Como se ha mencionado a través de todo el documento, lo que se ha realizado hasta ahora sólo ha quedado como prototipo. Sin embargo, podríamos decir que ya está todo el sistema listo para implementarlo y ponerlo en funcionamiento en una MMC real.

Las limitaciones son obvias y resultan ser del tipo mecánico. La siguiente etapa consistirá en diseñar el mecanismo de transmisión de movimiento en la MMC del Laboratorio de Metrología y acoplar el trabajo descrito aquí para poder finalmente integrar todas las partes en un instrumento de medición preciso. Asimismo, se deberá considerar la evaluación de un prototipo que realice las funciones propias de un palpador de contacto. Esto no se realizó debido a que implica un alto grado de complejidad del tipo mecánico y lo cual rebasa los alcances planteados en este trabajo. No obstante, cabe resaltar que nuestro prototipo emula tal dispositivo de una manera muy cercana a la real y que esto deja abierta la posibilidad de que en un futuro inmediato se realice.

El sistema electrónico, incluido el software, aún se puede mejorar y tal vez probar con otro tipo de tecnologías, pero en resumen, se necesitará ahora diseñar mecanismos de precisión que posicionen a la MMC en los niveles de exactitud deseados. Habiendo sorteado este último problema, estamos seguros que esta tecnología estará al alcance y será tan confiable como las MMC's actuales.

## ANEXOS

El presente capítulo muestra los diagramas diseñados para la elaboración del presente proyecto.

### A.1. Diagramas mecánicos

A continuación se muestran los diagramas mecánicos empleados para el prototipo. En todos los diseños se utilizó AutoCAD 2004.

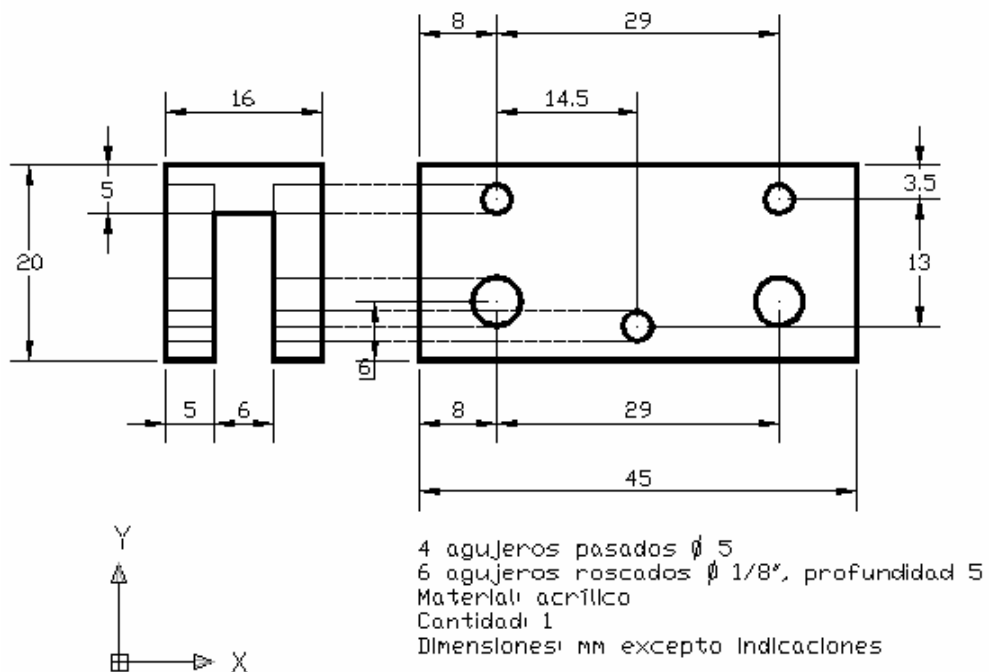
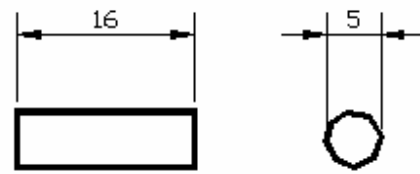
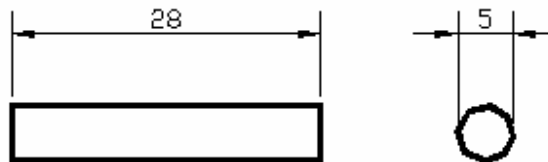
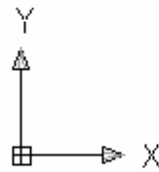


Figura A.1.1. Base.

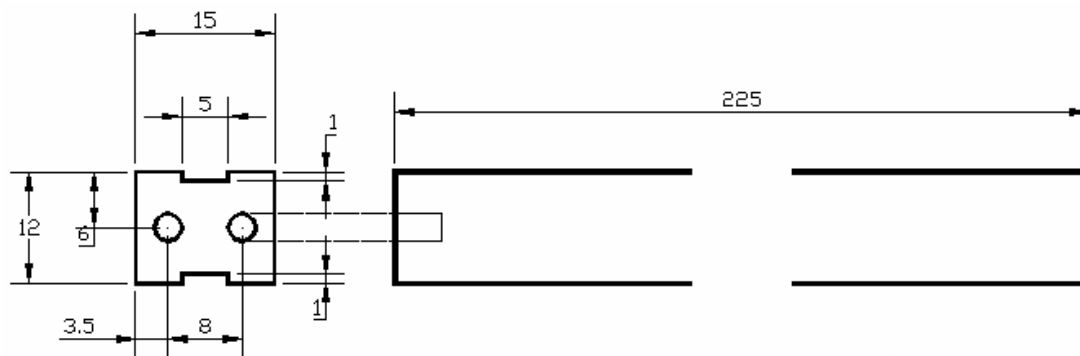


Material:acrílico  
Cantidad: 2  
Dimensiones: mm

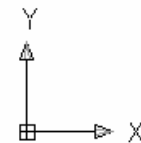


Material:acrílico  
Cantidad: 1  
Dimensiones: mm

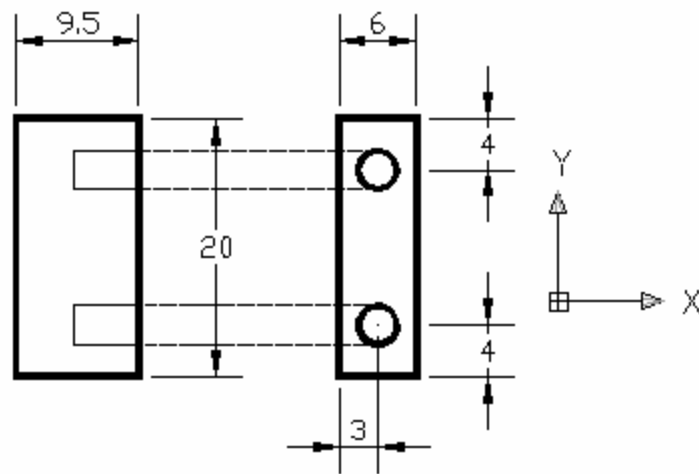
**Figura A.1.2.** Eje del balero.



4 agujeros roscados  $\varnothing 1/8"$ , profundidad 5  
Material:acrílico  
Cantidad: 1  
Dimensiones: mm excepto indicaciones

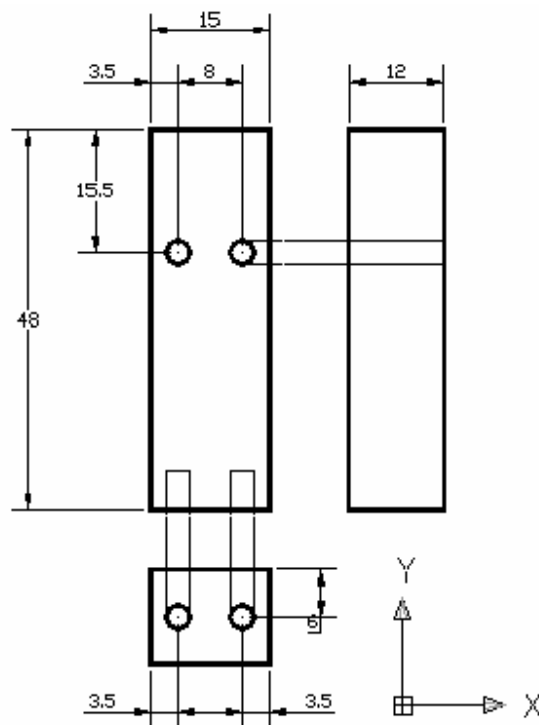


**Figura A.1.3.** Guía.



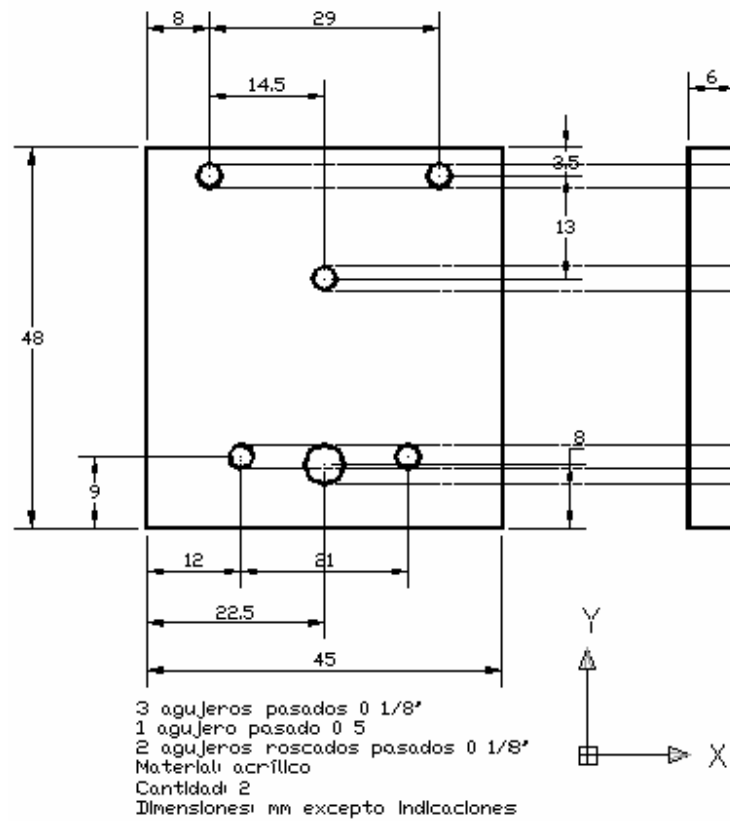
2 agujeros roscados  $\emptyset$  1/8", profundidad 5  
 Material: acrílico  
 Cantidad: 6  
 Dimensiones: mm excepto indicaciones

**Figura A.1.4.** Retén.

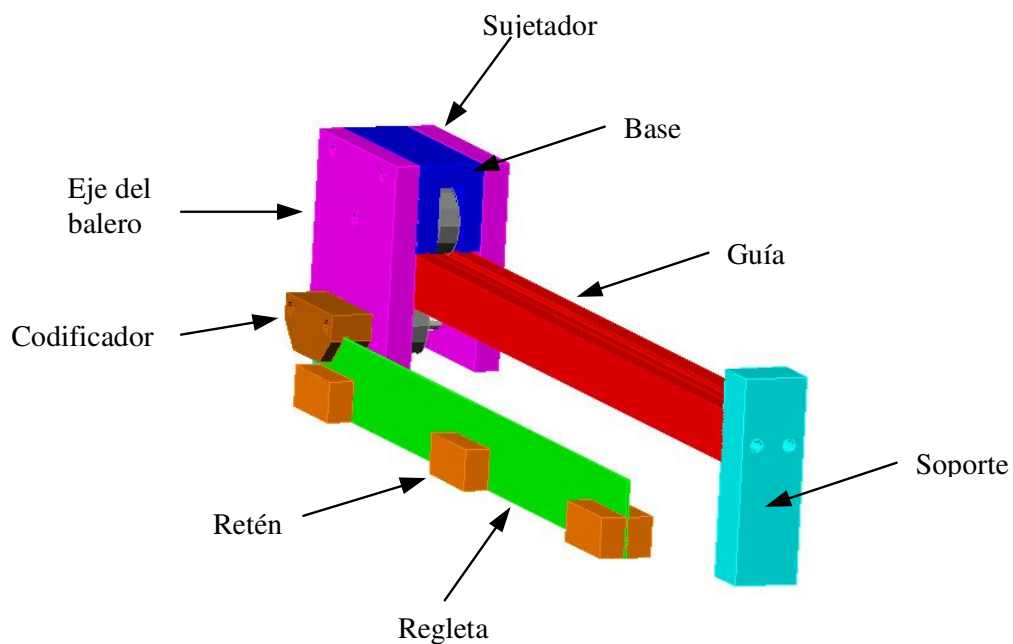


2 agujeros pasados  $\emptyset$  1/8"  
 2 agujeros roscados  $\emptyset$  1/8", profundidad 5  
 Material: acrílico  
 Cantidad: 2  
 Dimensiones: mm excepto indicaciones

**Figura A.1.5.** Soporte.



**Figura A.1.6.** Sujetador.



**Figura A.1.7.** Sistema completo para un eje coordenado.



## A.2. Diagramas electrónicos

Existen en general dos tipos de fuentes: las fijas y las variables [19]. Las primeras, como su nombre lo indica, son aquellas que sólo darán una tensión constante a la salida. Y las variables, en las que se puede regular la tensión dentro de un rango determinado en el diseño.

En nuestro caso, la fuente requerida fue una fija, ya que se necesitaban 12 volts para alimentar las tarjetas controladoras. Las figuras A.2.1 y A.2.2 muestran los diseños que se utilizaron para la elaboración de la fuente mediante el software ExpressPCB.

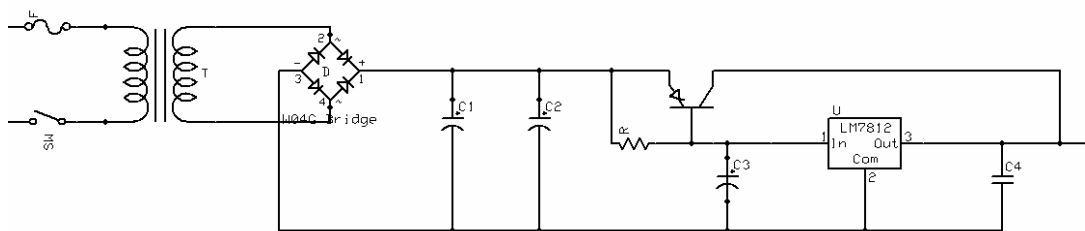


Figura A.2.1. Diagrama esquemático de la fuente.

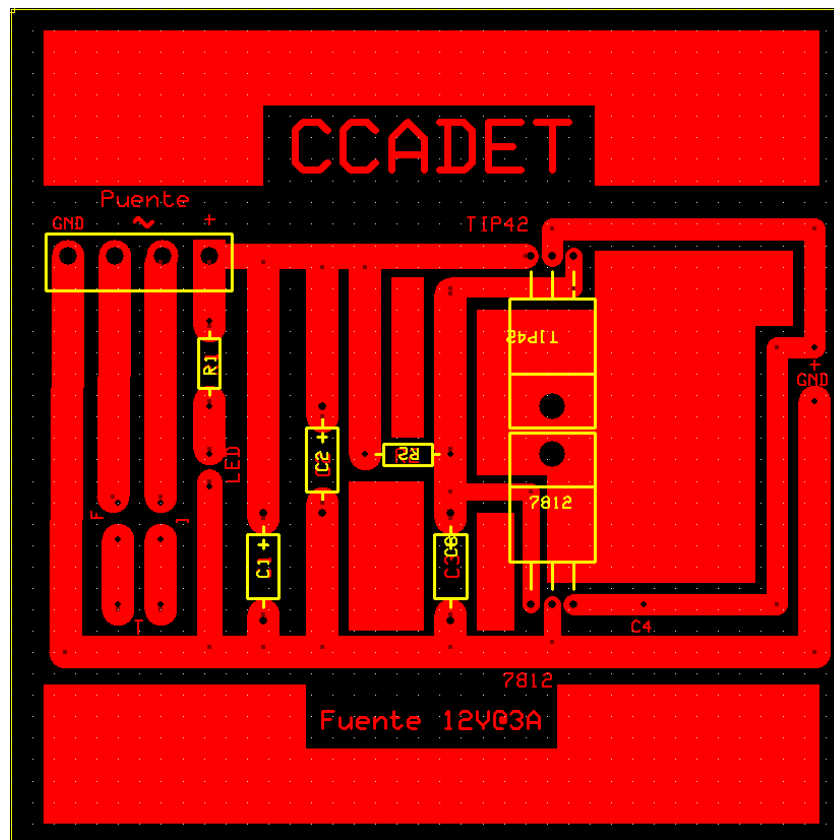


Figura A.2.2. Circuito impreso de la fuente.

Y cuyos materiales utilizados fueron:

- ✓ T: Transformador 12[V]@3[A]
- ✓ F: Fusible 3[A]
- ✓ SW: Interruptor
- ✓ Puente de diodos
- ✓ C1: Capacitor electrolítico 1000[ $\mu$ F]@50[V]
- ✓ C2: Capacitor electrolítico 10[ $\mu$ F]@25[V]
- ✓ C3: Capacitor electrolítico 1[ $\mu$ F]@50[V]
- ✓ C4: Capacitor electrolítico 0.1[ $\mu$ F]@50[V]
- ✓ TIP42: Transistor de potencia
- ✓ 7812: Regulador
- ✓ R: Resistencia 10[ $\Omega$ ]@ $\frac{1}{2}$ [W]
- ✓ Resistencia 330[ $\Omega$ ]@ $\frac{1}{2}$ [W]
- ✓ Leds
- ✓ Disipadores

### A.3. Diagramas de los conectores

La tabla A.3.1 muestra cómo se utilizó el cable DB15 con un código de colores para realizar las conexiones.

## Para el Conector 1

DB15	Descripción	Color
13	Tierra Codificador	Negro
5	Canal A Codificador	Amarillo
14	+5V Codificador	Rojo
6	Canal B Codificador	Azul
1, 2	M+ Motor	Azul, Amarillo
9, 10	M- Motor	Blanco, Verde
7	Limit Switch 2	Verde
4	Limit Switch 1	Rosa
12	GND	Café

Tabla A.3.1. Conector 1.

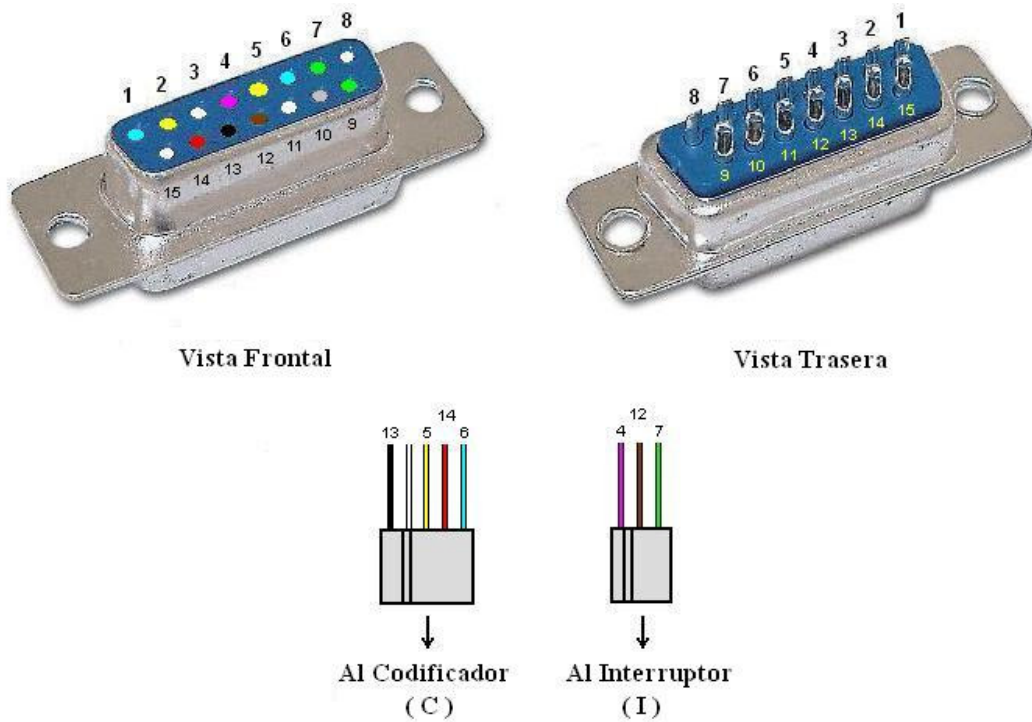


Figura A.3.1. Código de colores para el conector 1.

### Para los Conectores 2, 3

DB15	Descripción	Color
13	Tierra Codificador	Azul
5	Canal A Codificador	Amarillo
14	+5V Codificador	Rojo
6	Canal B Codificador	Negro
1, 2	M+ Motor	Amarillo, Negro
9, 10	M- Motor	Azul, Rojo
7	Limit Switch 2	Amarillo
4	Limit Switch 1	Rosa
12	GND	Negro

Tabla A.3.2. Conector 2 y 3.

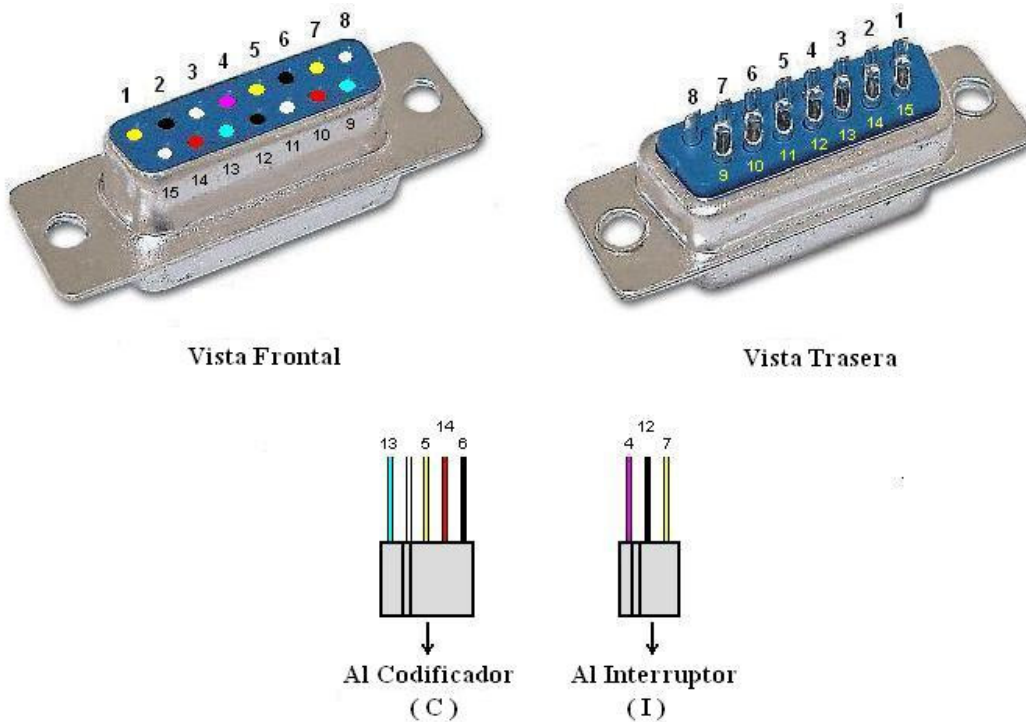


Figura A.3.2. Código de colores para el conector 2 y 3.

## REFERENCIAS

- [1]. S.G. Bermúdez, “Programación de la interfase de usuario entre una máquina de medición por coordenadas y una PC”. Ingeniería en Computación. Facultad de Ingeniería, UNAM. Tesis Licenciatura. Marzo de 2001. México, D.F.
- [2]. S.G. Bermúdez, B. Valera, J. Sánchez, R. Nava, “Interfase para una máquina de medir por coordenadas”, XIV Congreso de Instrumentación, SOMI. Tonantzintla, Puebla. Octubre. Pp 565-569.
- [3]. Centro Nacional de Metrología, CENAM, <http://www.cenam.mx/>, Marzo 2005.
- [4]. Comunicaciones, <http://www.automatas.org/abb/comunicaciones.htm>, Septiembre 2005.
- [5]. Control Tutorials for Matlab, <http://www.engin.umich.edu/group/ctm/>, Octubre 2005.
- [6]. Creaturoides, <http://www.creaturoides.com/anterior/srvesp.htm>, Octubre 2005.
- [7]. M. A. Cuevas, “Clasificación de las computadoras”, <http://www.monografias.com/trabajos11/comco/comco.shtml>, Septiembre 2005.
- [8]. D. J. Kruglinski, G. Shepherd, S. Wingo, “Programming Visual C++”, McGrawHill, 1153, 2000.
- [9]. Full Duplex – Wikipedia, [http://es.wikipedia.org/wiki/Full\\_Duplex](http://es.wikipedia.org/wiki/Full_Duplex), Septiembre 2005.
- [10]. J.R. Kerr, “PIC-SERVO Motor Control Board”, <http://www.jrkerr.com>, 5, (2003).

- [11]. J.R. Kerr, “Z232-485 Serial Port Converter”, <http://www.jrkerr.com>, 5, (2003).
- [12]. J. Mateos, “Prototipo de una Máquina de Medición de por Coordenadas”, <http://proton.ucting.udg.mx/expodiel/Enero93/PY27VI.html>, Septiembre 2005.
- [13]. D. Panello, “Yerba Mate y Visual C++”, <http://www.dcp.com.ar/>, Agosto 2005.
- [14]. Programación con Visual C++, <http://www.monografias.com/trabajos5/visualcurso/visualcurso.shtml>, Agosto 2005.
- [15]. Sensores Digitales, <http://www.investigacion.frc.utn.edu.ar/sensores/Tutorial/TECN O5.pdf>, Julio 2005.
- [16]. Sistemas para la adquisición de puntos en máquinas de medir por coordenadas, <http://www.metalunivers.com/Areas/metrologiadimensional/tutorial/sistemasadquisicion.htm>, Junio 2005.
- [17]. S. N. Spitz. A. J. Spyridi. A. A. G. Requicha, “Accessibility Analysis for Planning of Dimensional Inspection with Coordinate Measuring Machines”, <http://www-pal.usc.edu/papers/IRIS-98-360.pdf>, Agosto 2005.
- [18]. J. Tang, “PID Controller Using the TMS320C31 DSK with On-line Parameter Adjustment for Real-time DC Motor speed and Position Control”, *Proceedings of the IEEE International Symposium on Industrial Electronics*, (IEEE), 786-791, (2001).
- [19]. Tutoriales Informática – Fuentes de alimentación, <http://www.hispazone.com/conttuto.asp?IdTutorial=98>, Agosto 2005.
- [20]. Z. H. Xiong y Z. X. Li, “Error Compensation of Workpiece Localization”, *Proceedings of the 2001 IEEE International Conference on Robotics & Automation*, (IEEE), 2249-2254, (2001).
- [21]. Q. Yang, C. Butler, P. Baird, “Error Compensation of touch trigger probes”, *Measurement*, **18**, 1, 47-57, (1996).
- [22]. J. Zavaleta, “Control de movimiento para una máquina de medición por coordenadas”. Ingeniería en Computación. Facultad de Ingeniería, UNAM. Tesis Licenciatura. 6 de Diciembre de 2004. México, D.F.

- 
- [23]. J. Zavaleta, B. Valera, G. Ruiz, J. Sánchez, “Sistema para el control de movimiento en máquinas de múltiples ejes”, XVIII Congreso de Instrumentación, SOMI. México, D.F., a 6 de Octubre. pp ref. BVO1858.