Desarrollo e implantación de un control de velocidad para motores de corriente continua usando el microcontrolador LM629

Alberto Flores Rueda[1], José de Jesús Medel Juárez[2], Pedro Guevara López[3]

[1,2] Centro de Investigación en Computación
Instituto Politécnico Nacional
Av. Juan de Dios Batiz s/n C. P. 07738 México D. F., Tel. 57296000 Ext. 56523.
[1]aflores@pollux.cic.ipn.mx, [2]jjmedel@pollux.cic.ipn.mx, [3]pguevara@dfl.telmex.net.mx

Abstract: En este artículo describe en forma resumida el diseño y construcción de un sistema de control de velocidad aplicado a 2 motores de C. D (corriente directa). El proceso del control consta: *a) Un procesador anfitrión: Una PC, que maneja un sistema de control re-entrante para dos motores de C. D., independientes., b) Sistema de control que usa al dispositivo LM629, a través del cual se realiza el ajuste de la velocidad en cada motor hasta estabilizarlos, c) Un* puente "H" formado por el dispositivo LMD18201 para control de la corriente aplicada al motor seleccionado. En este trabajo se presenta: 1) Descripción del sistema desde dos puntos de vista: de hardware y del firmware, 2) La descripción de ambas estructuras se presenta a través de módulos. 3) La prueba de control de velocidad para dos motores por medio de la implantación de una ley de control ya dispuesta en el MPID,Los resultados permiten usar la tecnología re-entrante para controlar n máquinas con características similares pero con dinámicas acotadas dentro de vecindades de operatibilidad.

Palabras clave: Microcontrolador, implantación, sistemas de control, sistemas digitales.

1. Introducción

La automatización de **n** máquinas eléctricas rotativas requiere controlar la aceleración, la velocidad, y la posición de sus flechas [3]. El controlar estos tres estados dentro de esta clase de sistemas es algo que a través del uso de circuitos integrados tales como el LM629 y el LM18201 y un algoritmo de control preseleccionado en ellos, permitirá el desarrollo de un sistema de control re-entrante para **n** máquinas con dinámicas que se encontrarán en entornos cerrados y definidos. Para ello es necesaria la descripción del sistema de control (ver: [1]-[16]).

2. Descripción del Sistema

El proceso de control consta de un procesador anfitrión (una PC) que maneja una función re-entrante controladora de dos máquinas eléctricas. El sistema en hardware cuenta con dos subsistemas en que utilizan al dispositivo LM629, y al puente "H" LMD18201 para realizar el ajuste de la velocidad en cada motor a través del control de la corriente de sus devanados.

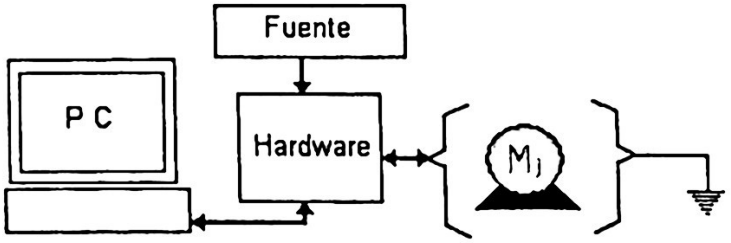

Fig. 1. Diagrama de un sistema de control para n máquinas a través de un solo sistema de hardware

2.1 Desde un punto de vista de Hardware

2.1.1 Descripción del sistema de control de velocidad

El sistema de control está definido por el procesador anfitrión estima y manda los comandos al microcontrolador LM629, en donde llega la señal observable del motor j-esimo y es decodificada dentro del MPID para actuar directamente en el filtro PID quien se encarga de corregir los valores que llegan al dispositivo de puente H y al circuito LM18201 (PH), que a su vez corrige la alimentación de corriente del motor.

El MPID, es un dispositivo dedicado para el control de motores de CD y servomecanismos usando las señales de retroalimentación de cuadratura incremental (ver la Fig. 2.) que da la flecha de la j-ésima máquina.

El LM629 tiene un bus periférico, y puede ser programado por una computadora personal, pero tiene preprogramado al filtro PID como compensador de lazo [2].

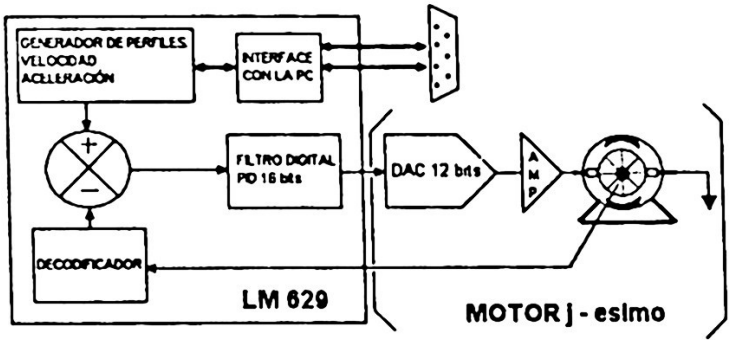

Fig.2 Sistema de control para el j – ésimo motor, basado en el LM629.

2.1.2 Descripción de la arquitectura del dispositivo MPID

En esta sección se detallan cada uno de los componentes y su uso dentro de la arquitectura del microcontrolador:

Se tiene una *ROM* de 1k x 16 bits que usa instrucciones de 16 bits, donde el algoritmo de control es almacenado. Se cuenta con un generador de secuencias *PLA* que decodifica esas instrucciones y da transferencia de señales de datos cronometradas para un bus de datos de 16 bits. Tiene una RAM donde son almacenados los parámetros de trayectoria de longitud doble (32 bits). El cual también es usado para valores de velocidad y aceleración. Y por último, tiene un ALU de 32 bits que se usa para soportar multiplicaciones de 16 x 16 bits para valores del error y coeficiente del filtro PID .

2.1.3 Descripción de la arquitectura del dispositivo puente H

El dispositivo de puente H [10][12], es de 3 amperes para aplicaciones de control de movimiento de motores de D.C. El puente H está construido en una estructura monolítica que incluye la multi-tecnología: *CMOS* para los circuitos de control, y *DMOS* para circuitos de potencia. La configuración de puente H es ideal para el manejo de motores de D.C. El puente H soporta picos de corriente de hasta 6 amperes.

2.2 Desde un punto de vista de Firmware

2.2.1 Módulos de programación

El microcontrolador maneja comandos e Instrucciones de dos bytes, un byte de 8 bits para la parte baja (desde el bit 0 hasta el bit 7) y un byte de 8 bits para la parte alta (desde el bit 8 hasta el bit 15), o sea una palabra de 16 bits[9].

El primer módulo que se requiere para una programación satisfactoria del MPID, es el módulo de prueba del *busy-Bit* : Coloca inmediatamente el bit cero en la parte baja del *byte de estado*, lo cual asegura satisfactoriamente la comunicación entre la computadora y el MPID después de que el procesador escribe un byte de comando, o lee o escribe el segundo byte del dato de la palabra (ver figura 3.). Cuando el *busy-Bit* es colocado, el MPID ignora cualquiera de los comandos o intentara transmitir un dato.

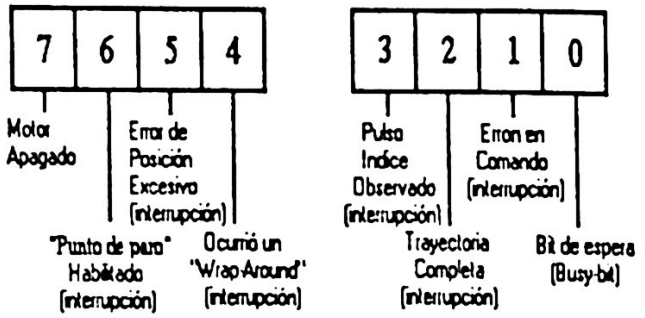

Fig. 3. Distribución de bits para interrupción del reset.

Leyendo el *Byte de estado* se cumple y se ejecuta un comando *RDSTAT*, el *RDSTAT* que es directamente soportado por el hardware del MPID, siendo este ejecutado para llevar /CS, /PS y /RD a un nivel lógico bajo.

2.2.2 Módulo de iniciación

En general, un módulo de iniciación contiene: Un comando de puesta a cero, otro de iniciación, un control de interrupción, y comandos de reportes de datos. Puede llevar el bloque de inicio para hardware y un comando PORT12.

Inmediatamente que entre la potencia, el inicio para hardware puede ser ejecutado, la puesta a cero del Busy-bit para hardware es iniciada para llevar la conexión */AST* (terminal 27), a un nivel lógico bajo. La rutina de inicio comienza después que */RST* es regresado a un nivel lógico alto, durante el tiempo de ejecución de la iniciación que es de *1.5 ms* máximo, el MPID ignora cualquier comando, o intentará transferir un dato.

2.2.3 Modulo de interrupciones

El comando *RSTI,* permite al usuario poner a cero los bits de la interrupción de bandera, unos seis bits de la palabra del byte de estado, como se puede ver en la Fig. 3., que contiene un comando *RSTI*, y una palabra de datos. El comando *RSTI,* inicia la puesta acero de los bits de la bandera de interrupción, y el comando RSTI también pone a cero el terminal de salida del interruptor de la computadora (terminal 17), enseguida del comando *RSTI*, una sola palabra de datos es escrita, el primer byte no es usado, y se usa el cambio de lógica cero en seis de los bits con lógica uno del segundo byte, iniciando las correspondientes interrupciones, como se ve en la Fig. 4. Cualquier combinación de los bits de la interrupción de bandera, serán puestos a cero en una sola secuencia del comando. Estas características permiten interrupciones que darán servicio de acuerdo a la prioridad de uso programado.

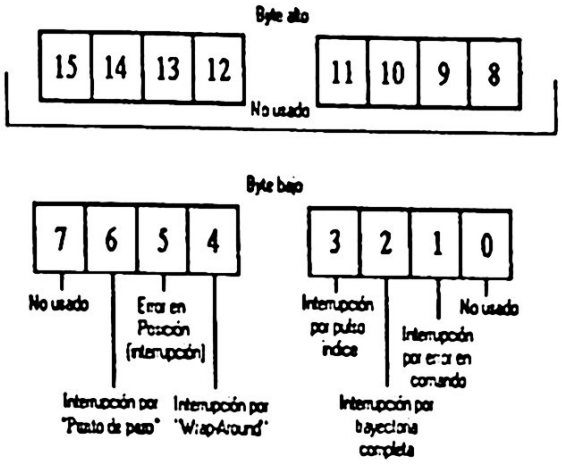

Fig. 4. Distribución de bits para interrupción del restaurador.

Consideraciones del restaurador del software.

El restaurador de software y hardware ejecutan los mismos trabajos, requiriendo el mismo tiempo de ejecución de *1.5 ms* como máximo. Durante la ejecución del restaurador del software, el MPID ignora cualquier comando, o intenta transferir un dato.

Módulo de programación del filtro

Una secuencia de un comando LFIL (carga de filtro) incluye: un comando LFIL, un control de filtro de la palabra, y un número variable de palabras de datos.

El comando LFIL inicia la carga del coeficiente del filtro, dentro de las entradas de los puertos

Los dos bytes de datos se escriben inmediatamente después del comando LFIL, que comprime la palabra del filtro de control. El primer byte del programa da el coeficiente de la muestra derivativa d_s (selección del intervalo de la muestra derivativa), y el segundo byte indica, con lógica de *unos* en las respectivas posiciones de los bits, en cuales está el remanente de cuatro coeficientes del filtro que pueden ser cargados.

Enseguida de la palabra del filtro de control, el coeficiente del filtro es escrito, cada coeficiente es escrito como un par de bytes de datos en una palabra, porque cualquier combinación de los cuatro coeficientes, puede ser cargada dentro de una sola secuencia del comando LFIL. El número de datos de las palabras, seguido de la palabra del control del filtro, puede variar en el rango de cero a cuatro.

Módulo de seguimiento de trayectoria

Una secuencia de un comando LTRJ (carga de trayectoria) incluye: El comando LTRJ, un control de trayectoria de la palabra, y una variable con un número de palabras de datos.
El comando LTRJ inicia cargando los parámetros de trayectoria en las entradas del buffer, y los 2 bytes de datos se escriben inmediatamente después del LRTJ, en los que se incluyen: la palabra de control de trayectoria, el primer byte del programa con lógica de unos en las respectivas posiciones del bit, el modo de la trayectoria (velocidad o posición), el modo de la dirección de la velocidad, y el modo de paro (ver módulo de paro), el segundo byte indica, con lógica de unos, las respectivas posiciones de cada bit, entonces los tres parámetros de trayectoria pueden ser cargados, también indica si los parámetros son absolutos o relativos, cualquier combinación de los tres parámetros puede ser cargada en una sola secuencia del comando LTRJ.

Seguido de la palabra de control de la trayectoria, el parámetro de la trayectoria es escrito, y cada parámetro es escrito como un par de palabras de datos (para bytes de datos). Para cualquier combinación, los tres parámetros podrán ser cargados en una sola secuencia del comando LTRJ.

Módulo de paro

Este módulo representa la forma requerida de pasar del estado de movimiento al estado de paro, porque el MPID opera en el modo de posición, normalmente el paro es siempre con desaceleración uniforme y ocurre automáticamente al final de una trayectoria especificada y en ocasiones es usado para desaceleración no uniforme para condiciones especiales, tales como el paro prematuro, etc.

Cuando el MPID opera en modo velocidad (considerando la desaceleración constante), el paro se realiza siempre por medio de este módulo.

Modulo de lectura para describir, posición, velocidad y aceleración

Este módulo consta de un programa maestro que utiliza los programas de los módulos de: inicialización, el del filtro, la secuencia del comando LTRJ y el comando STT describiendo los tres estados durante la trayectoria de la flecha.

Los factores que influyen el desarrollo de este programa son:
Un decodificador incremental de cuadratura, que codifica el eje de rotación en forma de pulsos eléctricos, en la Fig. 5., se detalla las señales generadas por un decodificador incremental de cuadratura de 3 canales M1, M2 e IN. El contador del MPID, decodifica la señal incremental de cuadratura para determinar la posición absoluta del eje.

La resolución de un decodificador de cuadratura integral [6], es usualmente especificado como el número de *líneas (éste* número indica el número de ciclos de la señal de salida) para cada revolución completa del eje de la flecha de la máquina eléctrica. Por ejemplo, un decodificador de *N líneas*, genera *N* ciclos de señales de salida, durante cada revolución.

Por definición, dos señales de dos canales A y B respectivamente, que están en cuadratura a 90°, o desfasadas 90° entre sí, se colocan juntas una de otra como se ve en la Fig. 5., se atraviesa por cuatro estados digitales distintos durante cada ciclo completo de cada canal. Cada transición de estado representa una *cuenta* de movimiento del eje, y el canal principal indica la dirección de rotación del eje.

Cada línea, sin embargo, representa un ciclo de las señales de salida, y cada ciclo representa *cuatro cuentas*, que se representan:

$$\left(N\frac{CICLOS}{REVOLUCION}\right)x\left(4\frac{CUENTAS}{CICLO}\right)=4N\frac{CUENTAS}{REVOLUCION} \qquad (1)$$

Si el sistema de referencia del codificador utiliza 1000 *líneas*, entonces:

$$\left(1000\frac{CICLOS}{REVOLUCION}\right)x\left(4\frac{CUENTAS}{CICLO}\right) = 4000\frac{CUENTAS}{REVOLUCION} \qquad (2)$$

Período de muestreo.

Considerando que el motor eléctrico gira a una velocidad constante, el muestreo que sucede de la posición actual del eje a una frecuencia fija, y es el recíproco del período de la muestra del sistema. Y el período de la muestra del sistema:

$$T_S = (2048)x\left(\frac{1}{T_{CLOCK}}\right) \qquad (3)$$

Período muestral del sistema.

Si el sistema de referencia usa un reloj de 8 MHz, entonces el período de la muestra del sistema de referencia que se sigue directamente al considerar a (3) es:

$$T_S = (2048)X\left(\frac{1}{8x10^6\ Hz}\right) = 256x10^{-6}\frac{s}{MUESTRA} \qquad (4)$$

Cálculo de parámetros de la trayectoria.

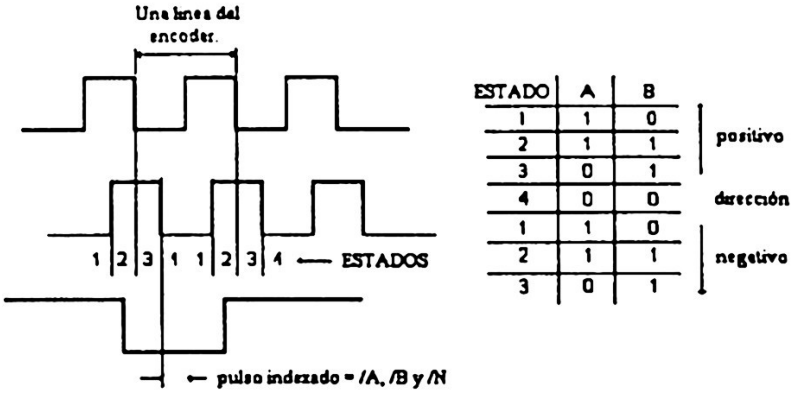

Figura 5. Señales generadas por un decodificador incremental de cuadratura de 3 canales.

Si el eje es acelerado a 0.1 rev/seg^2, hasta que alcance una velocidad máxima de 0.2 rev/seg, y entonces, desacelera hasta parar exactamente a dos revoluciones de la posición inicial Los cálculos de parámetros de la trayectoria para este movimiento son detallados enseguida.

NOTA: Una muestra en el tiempo se obtiene a través de pulsos en múltiplos de π radianes respecto al giro de la flecha (ver Fig. 6)

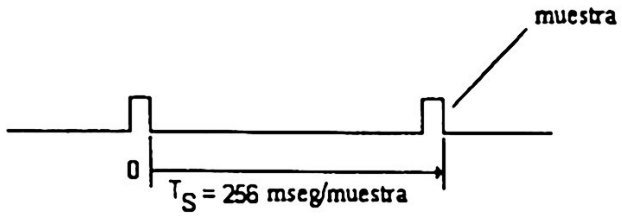

Fig. 6. Muestras obtenidas por impulsos.

Cálculo de velocidad, aceleración y posición.

A través de las expresiones (4) a (1), tenemos respecto a la aceleración

$$A = 00\ 00\ 00\ 02 \text{ hex} \qquad \frac{CUENTAS}{MUESTRA^2}$$

Ya que:

$$A = \left(4000 \frac{CUENTAS}{REVOLUCION}\right) x \left(256x10^{-6} \frac{SEGUNDOS}{MUESTRA}\right)^2 x \left(0.1 \frac{REVOLUCIONES}{SEGUNDO^2}\right) =$$

ada

$$A = 2.62x10^{-5} \frac{CUENTAS}{MUESTRA^2}$$

$$A = \left(2.62X10^{-5} \frac{CUENTAS}{MUESTRA^2}\right) x (65536) = 1.718 \frac{CUENTAS}{MUESTRA^2}$$

$$A = 2 \frac{CUENTAS}{MUESTRA^2}$$

De igual forma para calcular la velocidad

$$V = 00\ 00\ 34\ 6E \ \ hex \frac{CUENTAS}{MUESTRA}$$

Ya que:

$$V = \left(4000 \frac{CUENTAS}{REVOLUCION}\right) x \left(256x10^{-6} \frac{SEGUNDOS}{MUESTRA}\right) x \left(0.2 \frac{REVOLUCIONES}{SEGUNDO}\right)$$

$$V = 0.2048 \frac{CUENTAS}{MUESTRA}$$

$$V = \left(0.2048 \frac{CUENTAS}{MUESTRA}\right) x (65,536) = 13,421.77 \frac{CUENTAS}{MUESTRA} \quad Velocidad\ Escalada$$

$$V = 13,422 \frac{CUENTAS}{MUESTRA} \quad Velocidad\ redondeada$$

Y por último el cálculo de la posición lleva al siguiente resultado:

$$P = 00\ 00\ 1F\ 40 \ \ hex \ \ CUENTAS$$

Ya que:

$$P = \left(4000 \frac{CUENTAS}{REVOLUCION}\right) x (2\ 0\ REVOLUCIONES) = 8000 \ \ CUENTAS$$

3. Prueba del Sistema.

Prueba de motores bajo el programa de control 2.

Describimos a continuación el comportamiento de los motores para el perfil de velocidad que se muestra en la gráfica de la Fig. 7:

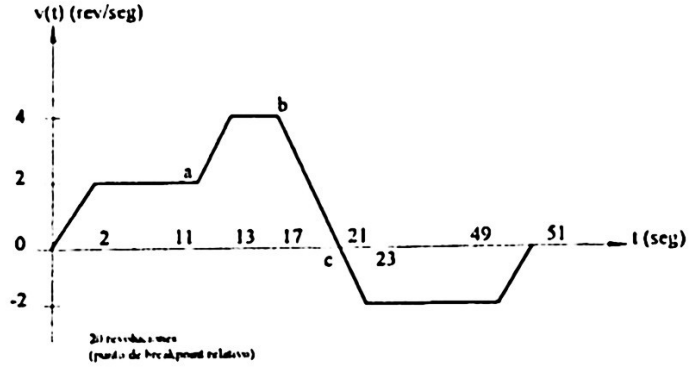

Fig. 7. (Perfil de velocidad para el control de trayectoria para ambos motores)

La flecha del motor acelera a 1rev/s^2 en una dirección hasta que alcance una velocidad constante máxima de 2 rev/sg. Después de completar 20 revoluciones en la misma dirección (incluyendo las revoluciones durante la aceleración) llega al punto ***a*** en la Fig. 7., entonces la flecha del motor se vuelve a acelerar a 1 rev/seg^2 para alcanzar una velocidad de 4 rev/seg. Después de completar 20 revoluciones en esa dirección (incluyendo las revoluciones durante la aceleración) y llegar al punto ***b*** en la Fig. 7., el motor desacelera (a 1 rev/seg^2) hasta parar. Finalmente, a partir del punto ***c*** de la Fig. 7., la flecha del motor acelera a 1 rev/seg^2 para alcanzar una velocidad de 2 rev/seg, y después de completar un número de revoluciones desacelera (a 1 rev/seg^2) y para, pero esta vez en dirección inversa.
La programación del MPID para lograr el perfil de velocidad de la gráfica de la Fig. 7., se describe a continuación:

Se inicia programando al MPID en *modo velocidad.* Se habilita un punto de paro relativo en el punto ***a*** de la Fig. 7. Esto representa un movimiento de 20 revoluciones en una dirección con una velocidad máxima de 2 rev/seg, esto es, tardará 10 segundos en cubrirlas. Cuando esta posición es alcanzada, el MPID manda una interrupción a la PC, y el ordenador anfitrión ejecuta una secuencia de comandos que incrementan la velocidad, además deshabilita la bandera de interrupción para el punto de paro, y carga un punto de paro absoluto. Es importante notar que en esta parte el MPID se programa para recibir un valor de velocidad relativo. El valor de velocidad que se carga es de 2 rev/seg, con lo cual el valor final de velocidad será de 4 rev/seg.

El punto de paro absoluto se representa como el punto ***b*** en la Fig 7. Cuando esta posición es alcanzada, el MPID interrumpe al ordenador anfitrión, y éste ejecuta un módulo de paro paulatino o desaceleración exponencial.

En el punto ***c*** de la gráfica el MPID es reprogramado, ahora en *modo de posición relativa.* Con este modo se asegura que, independientemente de la posición restante de la flecha, ésta completará treinta revoluciones en la dirección inversa, es decir en 15 seg.

En la Fig. 7., se observa que podemos tener diferentes velocidades de acuerdo a nuestro control, y también tener diferentes aceleraciones para poder alcanzar en un cierto tiempo esas velocidades. Para poder observar el cambio de velocidad y aceleración, se manejan los motores a baja velocidad y aceleración; para precisar estos cambios, nuestra velocidad fluctúa de 1 a 15 rev/seg en los valores de velocidad y de 1 a 15 rev/seg2 en los valores de aceleración.

4. Conclusiones

En este artículo describo en forma resumida el diseño y construcción de un sistema de control de velocidad aplicado a 2 motores de C. D (corriente directa). El proceso del control esta formado por:

a) Un procesador anfitrión: Una PC, que maneja un sistema de control re-entrante para dos motores de C. D., independientes.,
b) Sistema de control que usa al dispositivo LM629, a través del cual se realizó el ajuste de la velocidad en cada motor hasta estabilizarlos,
c) Un puente "H" formado por el dispositivo LMD18201 a través del cual se controló la corriente aplicada al motor seleccionado.

En este trabajo se presentó:
1) Descripción del sistema desde dos puntos de vista: de hardware y del firmware,
2) La descripción de ambas estructuras a través de módulos.
3) La prueba de control de velocidad para dos motores por medio de la implantación de una ley de control ya dispuesta en el MPID,

Los resultados muestran que el usar la tecnología re-entrante para controlar **n** máquinas con características similares es factible con la combinación PC – MPID, considerando dinámicas acotadas dentro de vecindades de control.

Referencias

[1] **Jhon Webb & Kevin Greshock.** Industrial Control Electronic Ed. Merrill 1997.

[2] **Curtis D. Johnson.** Process Control Instrumentation Technology. Ed. John Wilex & Sons. 1995

[3] **W.F. Stoccker & P.A. Stoccker.** Microcomputer Control of Thermal and Mechanical System. University of Illinois at Urbana-Champaign. Hewlett-Packard Company. Ed. Van Nostrand Reinhold New York. 1989.

[4] **M.Gopal.** Digital Control Enginnering. Electrical Engineering, Department Indian Institute of Technology Delhi India. Ed. Jhon Wilex and Sons.1992

[5] **Kenneth** Microcontroler Ed. Mc Graw-Hill International Editions 1996.

[6] **National Semiconductor.** Aplications Hand Book, AN-706. 1995 y 1993.

[7] **National Semiconductor.** LM628/LM629 Precision Motion Controler February 1995

[8] **Robert L. Hoekstra.** Robotics and Automated System. Ed. South-Western Publishing Co.1997

[9] **National Semiconductor.** LM628/LM629 Programming Guide, (nota 693).1998

[10] **National Semiconductor.** LMD18201 April 1998

[11] **Denis o´Kelly.** Performance and control of Electrical Machines. Ed Mc Graw Hill 1992

[12] **Philip T. Krein.** Elements of Power Electronics Ed. Oxford University Press. 1998.

[13] **Milos D. Ercegovac & Tomás Lang.** Digital Sistem and Hardware/Firmware Algorithms

[14] **Paul M. Embree &Bruce Kimble.** C Language Algoritms for Digital Signal Processing. Ed. Prentice Hall. 1991.

[15] **Texas Instrument.** TTL Logic Data Book. 1988.

[16] **CRC PRESS & IEEE PRESS.** The Control handBook. Ed. Williams Levine. 1996.