

# DESARROLLO DE UN AMBIENTE DOCENTE E INVESTIGATIVO EN SERVOCONTROL DIGITAL PARA ROBÓTICA EXPERIMENTAL

Roberto S. Apóstoli (\*) (\*\*); Alfio Mármol (#); Gerardo Artola (\*)  
(\*)Lab.Proc.Láser-Ciop-Conicet, (\*\*) Sec.Cienc.y Tec. Fac. Reg. Bs. As. Univ.Tec.Nac.  
Filemón Posse 2736. (5009) Córdoba [rsapostoli@arnet.com.ar](mailto:rsapostoli@arnet.com.ar)  
(#) Fac. Cienc. Exact. Fís.y Nat. Universidad Nacional de Córdoba.  
Montevideo1229 B° Res.El Crucero .(5186). Alta Gracia.Cba. [alfiomarmol@yahoo.com.ar](mailto:alfiomarmol@yahoo.com.ar)  
(\*)Lab.Proc.Láser-Ciop-Conicet  
Huartes 2428. B° Los Paraísos (5000) Córdoba [gerartola@yahoo.com](mailto:gerartola@yahoo.com)

## Resumen

Se describe el desarrollo de un ambiente para la enseñanza y la investigación sobre servomecanismos digitales aplicados a la robótica y al control industrial experimental.

Se expone la elección del controlador y los diagramas en bloque y funcional del sistema expresado en transformadas de Laplace (L) y Z. Se detalla la implementación tecnológica junto al programa en Visual Basic interactivo para el usuario conducente a obtener un banco de ensayos de laboratorio, y se presenta la metodología de diseño del filtro y del control digital junto con las simulaciones del diagrama de Bode y las transformadas de L y Z en Matlab y Simulink.

En el proyecto se aplicó fundamentalmente ingeniería nativa multidisciplinaria: informática, electrónica y mecánica. El sistema se encuentra operativo comprobándose con su funcionamiento en tiempo real en el banco de ensayos y en el robot RSA 2 los algoritmos y simulaciones implementados en una PC estándar.

## Palabras Claves

Informática educativa. Tratamiento de señales. Algoritmos. Arquitectura de procesamiento.

## Descripción General

Previa a la decisión de elegir un componente específico para la implementación del proyecto, se estudió en el mercado de EEUU cuáles eran las placas y chips de control disponibles para los servomotores de C.C., “brushless” y step-motor. En cuanto a las placas, las que ofrecen las empresas Galil, Acroloop, Aerotech y National Instruments son las más adecuadas. Su precio FOB arranca desde unos 700 u\$s para controlar dos ejes con un tiempo de muestreo de 256 micro segundos. Para controlar 4 ejes con bus PCI y con un tiempo de muestreo de 64 microseg, la placa cuesta unos 1600 u\$s FOB. Por razones estratégicas y de costo se resolvió diseñar una placa de control digital basada en un chip dedicado. Se preseleccionaron dos de los más conocidos: el National LM628/629 y el Hewlett-Packard (HP) HCTL 1100. Ambos cuestan en EEUU unos 48 u\$s c/u. aprox. La siguiente tabla expone sus características relativas:

<i>Tipo de chip:</i>	<i>Tiempo de muestreo en microsegundos.</i>	<i>Resolución en bits</i>	<i>Control de motores</i>		
			<i>C.C. “brushless”</i>	<i>Step</i>	<i>Step</i>
LM 628 /629 a 8Mhz.	256	32	Sí	Nó	Nó
HTCL 1100 a 2 Mhz.	64	24	Sí	Sí	Sí

Se decidió por el HCTL 1100 por su menor tiempo de muestreo y la posibilidad de controlar los tres tipos de motores. Se diseñó una arquitectura de control como se muestra sintéticamente en la figura 1.

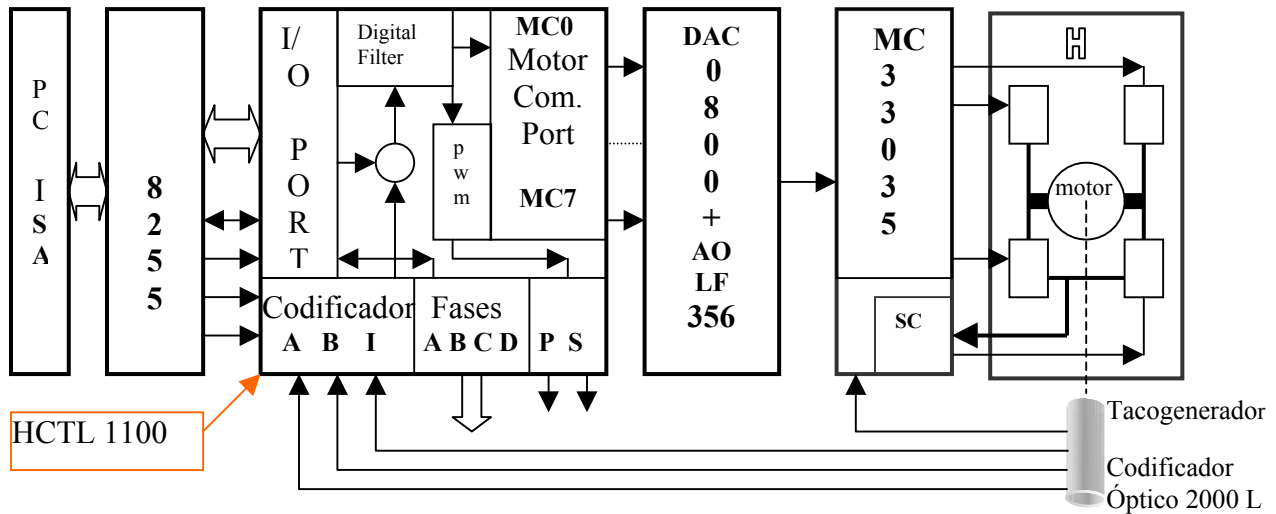


Fig.1 Diagrama en bloques del control digital de servomotores

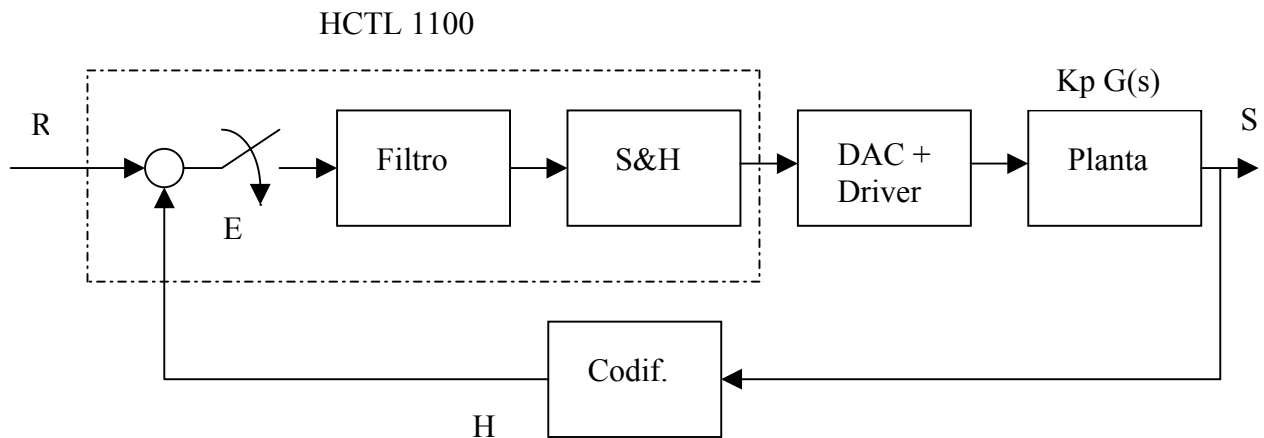


Fig.2 Diagrama funcional del control digital

En la figura 2 se expone el diagrama funcional del control digital para su análisis en el campo discreto. El HCTL 1100 posee internamente un filtro digital del tipo de adelanto de fase (aportado mayoritariamente por el cero) con la transferencia en transformada Z:

$$Gf(z) = K \frac{(z - A)}{(z + B)} \quad (1) \quad \text{Este filtro se denomina FIR (Finite Impulse Response). Sus ventajas son:}$$

- Menores errores de propagación al ser no recursivo.
- Realización sencilla por convolución (aplicando transformada rápida de Fourier FFT)
- Mayor estabilidad porque sus polos están en el origen.

También internamente el chip posee un retenedor de orden cero ROC (ZOH zero order hold, S&H sample and hold ).

Su transferencia en transformada de Laplace  $L [Gsh]$  es: 
$$Gsh(s) = \frac{1 - e^{-sT}}{s} \quad (2)$$

La constante T es el tiempo de muestreo del control digital. En nuestro caso el valor mínimo en el HCTL 1100 es de 64  $\mu$ seg.

Las transferencias del conversor analógico-digital DAC y del driver de potencia (que excita al servomotor) se pueden expresar cada una con una ganancia ó constante aperiódica.

La planta la constituye el servomotor con su carga mecánica asociada y su transferencia es  $KpG(s)$ . Aplicando la transformada de Laplace  $L$  y la transformada zeta  $Z$  al retardo de la respuesta del retenedor de orden cero ROC se puede (agrupándolo conjuntamente con el conversor DAC, el driver y la planta) expresar con una transferencia equivalente a:

$$G(z) = (1 - z^{-1}) Z \left[ \frac{KeG(s)}{s} \right] \quad (3)$$

Combinando en cascada las expresiones (1) y (3) se obtiene la transferencia en  $Z$  del sistema abierto entre la salida S y el error E. La salida S es una posición mecánica dada en radianes.

El codificador óptico de 2000 líneas/vuelta provee los pulsos en los canales A, B (desplazado 90°) e índice I (un pulso/vuelta). Su transferencia H es una constante dada en pulsos por radianes. El codificador se conecta a través de un línea RS 422 (de baja impedancia inmune al ruido) mediante una dupla constituida por dos "line drivers": DS26LS31 y DS26LS32 que operan en modo de transmisión por corriente.

## Implementación del proyecto

En la figura 3 se presentan las fotografías del proyecto realizado. Se construyó un banco de ensayos constituido fundamentalmente por el conjunto principal de una impresora de alta calidad Wang en desuso la cual posee en su chasis un servomotor de muy baja inercia Kollmorgen con inducido de circuito impreso y tacogenerador analógico y una guía lineal sobre rodamientos para soportar ahora una carga variable de ensayos (anteriormente era el inyector de tinta original). Se agregaron el codificador óptico y el driver de potencia para el servomotor.

En el proyecto se ha desarrollado el driver en base al chip de Motorola MC33035, el cual maneja directamente la llave H. Este chip originalmente fué previsto para excitar motores de C.C. sin escobillas ("brushless") de 3 fases pero se lo ha adaptado para servomotores estandard de C.C. El chip incluye un generador PWM con reloj de 20 Khz, un AO amplificador operacional para recibir la señal de comando y otro amplificador para controlar la corriente máxima. Se agregaron tres AO LF356 para acondicionar la realimentación del tacogenerador y la ganancia del driver. La llave H se implementó en la rama superior con dos mosfet IRF 640 de 125 watts c/u. En la rama inferior se utilizaron dos IRC 640 de 5 pines que entregan una señal proporcional (método Kelvin) a la corriente de Drain. El driver puede entregar 18 amp. continuos y 72 amp. pico a 120 volts de c.c.

Se desarrollaron las placas de los circuitos impresos utilizando el software "Protel 98" y se ejecutaron en CITEM SRL. con tecnología "through hole". Protel 98 permite realizar una gran diversidad de circuitos impresos (simple y doble faz) con o sin "through hole". Para ello primero se realizó en Protel 98 el circuito esquemático con todos los componentes y conexiones (archivo con extensión .sch).

Luego se realiza el archivo con extensión .pcb en el cual se establecen las dimensiones de la placa, la ubicación de cada componente y las condiciones de ruteo de pistas (separación entre pistas, ancho, etc.), para que el software determine como trazar las pistas de la mejor manera, esto es, trazar pistas lo más cortas y con menos saltos posibles. Con este archivo tipo .pcb el fabricante desarrolla la placa. Se exponen en la figura 4 a y 4 b el circuito impreso de la placa Bus ISA para el controlador digital HCTL 1100.

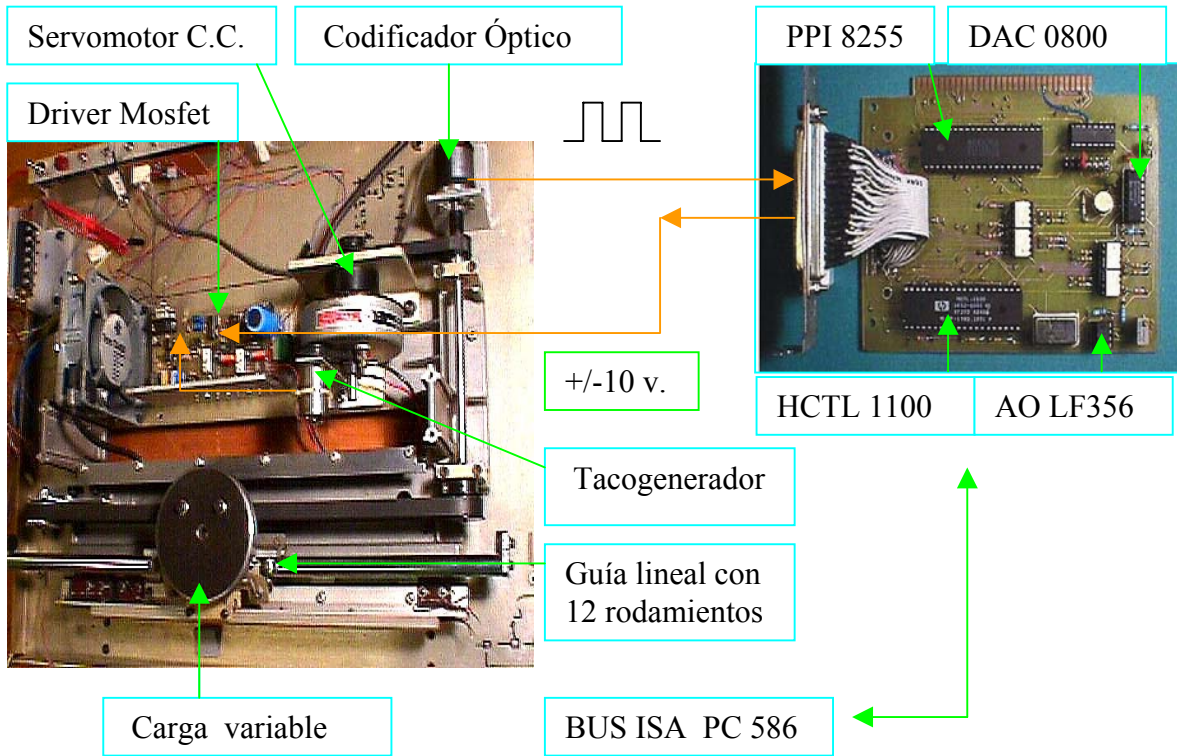


Fig. 3 Layout del Banco de ensayos para control digital

La señal analógica de comando de +/- 10 volts, que proviene del circuito impreso, la entrega un amplificador operacional AO LF 356 acondicionando la señal del conversor Digital Analógico de 8 bits DAC 0800, el cuál toma los datos digitales del Microcontrolador Digital de Motores HCTL 1100. Este circuito integrado CI es un banco de 64 registros de 8 bits, donde sólo 35 pueden ser utilizados por la PC (Host) y el usuario. Posee un bus de Direcciones/Datos bidireccional que permite escribir y leer dichos registros y con ello establecer las condiciones para controlar el motor. Tiene también tres canales para tomar las señales del encoder acoplado al motor y cerrar de esta manera el lazo de realimentación. Es decir, lee permanentemente en que posición está el motor, y con ello su velocidad y aceleración. Esta lectura en cuadratura de los pulsos del encoder es actualizada constantemente en tres registros, permitiendo leerla en cualquier momento y actuar en consecuencia. A su vez presenta cuatro líneas con las cuales se controla el flujo de información, es decir, con ellas se establece si lo que está en el bus de Direcciones/Datos es un Dato o una Dirección, o si se trata de una operación de Lectura o Escritura. En definitiva, solo hay que realizar operaciones de escritura y lectura en este CI.

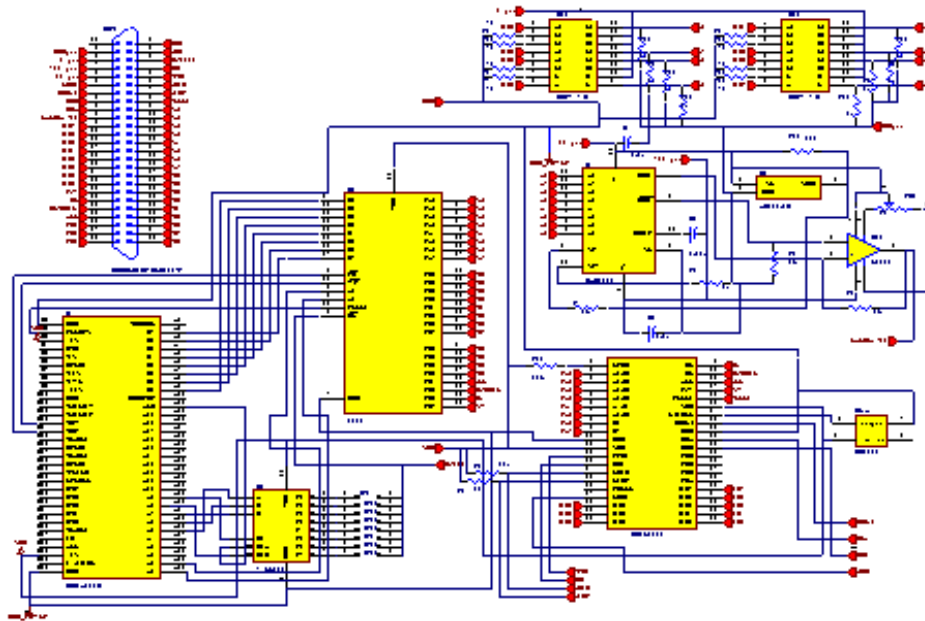


Fig. 4a Dibujo esquemático .sch

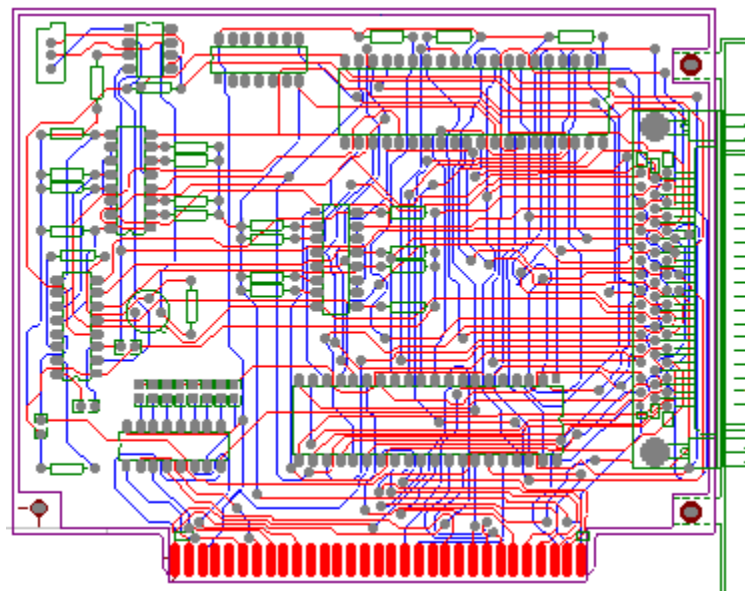


Fig.4b Circuito impreso .pcb

Se utiliza una PPI (Programmable Peripheral Interface) 8255 que posee tres puertos paralelos de 8 bits con direccionabilidad programable. En el proyecto se aprovechan el Puerto A para el Bus de Direcciones/Datos y cuatro bits del Puerto C para las líneas de control del HCTL 1100. Si bien el 8255 se puede manejar a través de varios Lenguajes de Programación, en este caso se aplicaron el QBASIC, Visual BASIC y Visual C++. Finalmente se decidió realizarlo en Visual BASIC por su simplicidad para programar y por las prestaciones que un entorno visual ofrece. Visual BASIC no posee instrucciones para lectura y escritura de puertos de E/S, por lo cual se tuvo que desarrollar una librería DLL (Dynamically Linked Libraries) en Visual C++ y llamada 8255.dll.

Para ello primero hay que crear un archivo 8255.def que contenga lo sig:

---

```
LIBRARY 8255
EXPORTS
    Out8255            @1
    In8255             @2
```

---

Luego hay que crear también un archivo llamado 8255.cpp que contenga lo sig:

---

```
#include
#include // librería que contiene las funciones _inp y _outp de Visual C++

short _stdcall Out8255 ( int PortAddress, int PortData )
{
    short Dummy;
    Dummy = (short) (_outp (PortAddress, PortData ));
    return (Dummy);
};
short _stdcall In8255 ( int PortAddress )
{
    short PortData;
    PortData = (short) (_inp (PortAddress ));
    return (PortData);
}; // Necesitamos Dummy por que _outp retorna un entero (int) el cual es de 32 bits
// en Win32 C++, mientras que un entero corto (short) es de 16 bits en Win32 C++.
// Por lo tanto, utilizamos (short) para forzar que C++ nos retorne un entero de 16 bits
// a Visual Basic.
```

---

Una vez creados estos archivos en una carpeta separada, se va a Visual C++, creando un nuevo proyecto Win 32 Dynamic-Link-Library, estableciendo una ubicación y colocándole un nombre (8255). Luego se agregan al proyecto los archivos realizados. Y por último se selecciona en el Menú "Build" el comando "Build 8255.dll". Con lo cual se crea la librería que se necesita.

En dicha librería se han creado dos funciones **In8255** y **Out8255**

Para utilizar dichas funciones hay que definir la librería en el encabezado del programa en Visual BASIC, como sigue:

**Option Explicit**

**Private Declare Function Out8255 Lib "8255.dll" (ByVal PAdd As Integer, ByVal PData As Integer) As Integer**

**Private Declare Function In8255 Lib "8255.dll" (ByVal PAdd As Integer) As Integer**

A continuación se muestra la secuencia de instrucciones utilizadas para leer y escribir un registro del HCTL 1100 con Visual Basic.

<b>Escribir</b>	<b>Leer</b>
Dummy = Out8255(Control, 128)	Dummy = Out8255(Control, 128)
Dummy = Out8255(PuertoA, 0)	Dummy = Out8255(PuertoA, 0)
Dummy = Out8255(PuertoC, 96)	Dummy = Out8255(PuertoC, 96)
Dummy = Out8255(PuertoC, 112)	Dummy = Out8255(PuertoC, 112)
Dummy = Out8255(PuertoC, 80)	Dummy = Out8255(Control, 144)
Dummy = Out8255(PuertoA, 11)	Dummy = Out8255(PuertoC, 208)
Dummy = Out8255(PuertoC, 112)	Dummy = Out8255(PuertoC, 240)
	Dummy = Out8255(PuertoC, 48)
	VALOR = In8255(PuertoA)
	Dummy = Out8255(PuertoC, 112)

Donde Control, PuertoA y PuertoC son constantes predefinidas que dependerán de la dirección del puerto de E/S que se elige para que la CPU encuentre la placa colocada en el Bus ISA.

En la placa ISA se coloca una serie de Jumpers con el fin de poder configurar el Hardware con varias direcciones ( 200H, 220H, 240H, 260H, 280H, 2A0H, 2C0H, 2E0H).

La dirección de: "Control" será la dirección de la placa +3 (en el proyecto: 283H ).

"PuertoA" será la dirección de la placa (en el proyecto: 280H ).

"PuertoC" será la dirección de la placa +2 (en el proyecto: 282H ).

La ventana principal del Software desarrollado en Visual Basic se muestra en la figura 5.

### **Metodología para el diseño del Filtro Digital**

Para la modelización del sistema hay que obtener la función de transferencia de cada elemento del sistema. Se expondrá a continuación solamente la función de transferencia del Motor + Carga.

La función de transferencia de un motor controlado por tensión es:

$$\frac{\theta_m(s)}{V_a(s)} = \frac{K}{(sT_e + 1)(sT_m + 1)} \quad (4) \quad ; \text{donde } K = \frac{K_c}{f \cdot r + K_v \cdot K_c} \quad ; T_m = \frac{J \cdot R}{f \cdot r + K_v \cdot K_c} \quad \text{y} \quad T_e = \frac{L}{R}$$

Reemplazando con los valores de catálogo del motor Kollmorgen, la función de transferencia es: Fórm. (5).

En la Figura 6 se muestra el diagrama de bloques del sistema que se desea compensar, simulándolo en el programa Simulink y Bode de Matlab 5.2. Como se aprecia en la Figura 7 y en la Figura 9 el sistema es inestable.

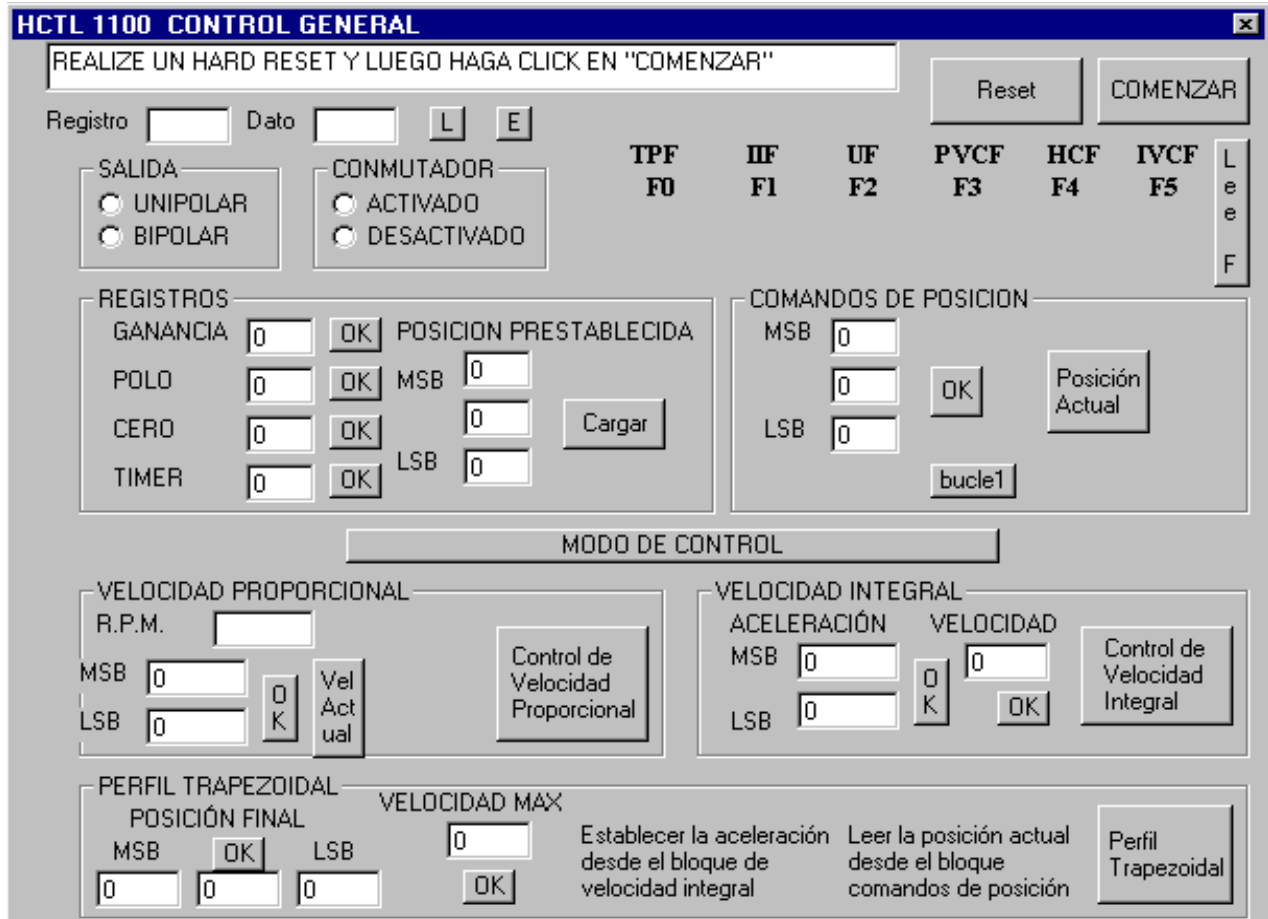


Fig. 5 Pantalla interactiva para el usuario

$$\frac{\theta_m(s)}{V_a(s)} = \frac{5}{(s1 \times 10^{-4} + 1) \cdot (s19 \times 10^{-3} + 1)} \left[ \frac{\frac{N.cm}{Amp}}{\left( \frac{N.cm}{kRPM} \cdot \Omega \right) + \left( \frac{N.cm}{Amp} \cdot \frac{V}{kRPM} \right)} \right]$$

$$\frac{\theta_m(s)}{V_a(s)} = \frac{27.56}{(0.0000019s^2 + 0.0191s + 1)} \left[ \frac{rad}{V \cdot seg} \right] \quad (5)$$

Para compensar el sistema con una respuesta al escalón, como la mostrada en la Figura 8 se agrega al sistema el filtro digital del HCTL 1100. (ver Figura 10)

El filtro se desglosa en tres factores.

$$D(z) = \frac{[K] \cdot [z - A]}{[4] \cdot [z + B]} = \left[ \frac{K}{4} \right] \cdot \left[ \frac{z - A}{z} \right] \cdot \left[ \frac{z}{z + B} \right]$$



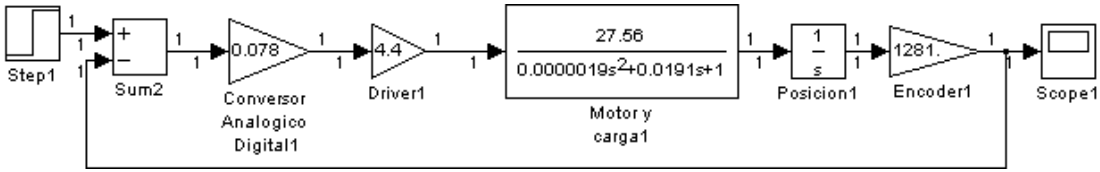


Fig. 6 Diagrama de bloques del sistema a compensar

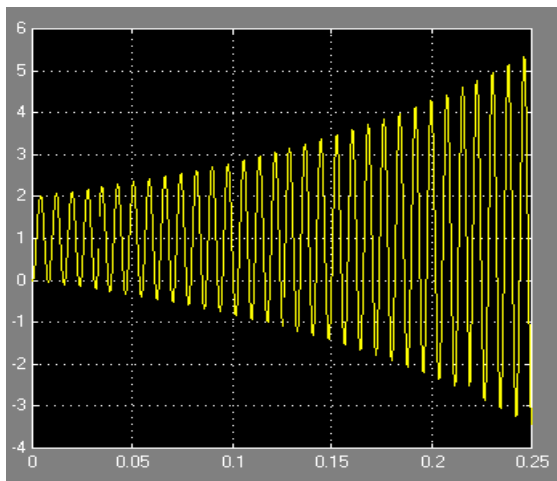


Fig.7 Respuesta del sistema sin compensar al escalón unitario.

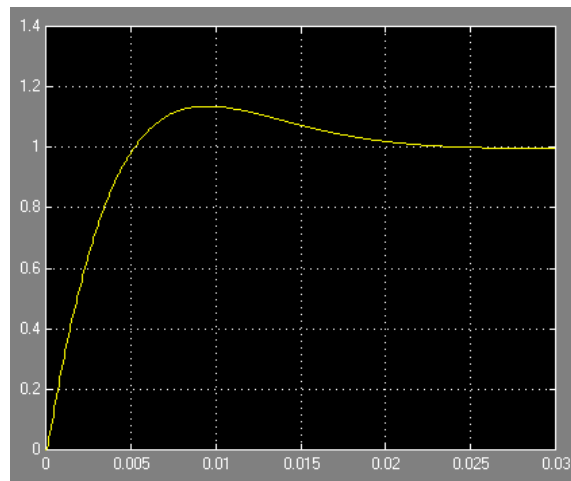


Fig. 8 Respuesta del sistema (compensado con filtro digital ) al escalón unitario.

Luego se transforma cada parte en Laplace y se analiza la respuesta en frecuencia de cada uno para distintos valores de ceros y polos. Es decir, magnitud y fase para distintos A y B, con frecuencia normalizada  $\omega_N = \omega \cdot t$ , donde t es el tiempo de muestreo elegido.

Vemos que el polo (B) y el cero (A) contribuyen ambas, a un adelanto de fase del sistema. El termino K es un factor de ganancia independiente que puede ser usado para levantar la ganancia del sistema y lograr el mayor ancho de banda, y para compensar la reducción de ganancia que introducen el polo y el cero del filtro digital. Siguiendo a Hewlett-Packard se simplifica la expresión del ROC aproximándola a una exponencial, modalidad adoptada por varias industrias.

Con las gráficas 11 y 12 se realiza el cálculo de los parámetros del filtro en función del Ancho de Banda y el Margen de Fase deseado.

Se compensa para un Margen de Fase entre  $60^\circ$  y  $70^\circ$ , por ej.  $65^\circ$  y para un BW de 60 Hz.

A las gráficas se ingresa con la frecuencia normalizada, se elige  $t = 64 \mu\text{seg}$ .

$$\text{En consecuencia } \omega_{NC}' = \omega_C' \cdot t \rightarrow \omega_{NC}' = 377 \frac{\text{rad}}{\text{seg}} \cdot 64 \mu\text{seg} \Rightarrow \omega_{NC}' = 0.024128$$

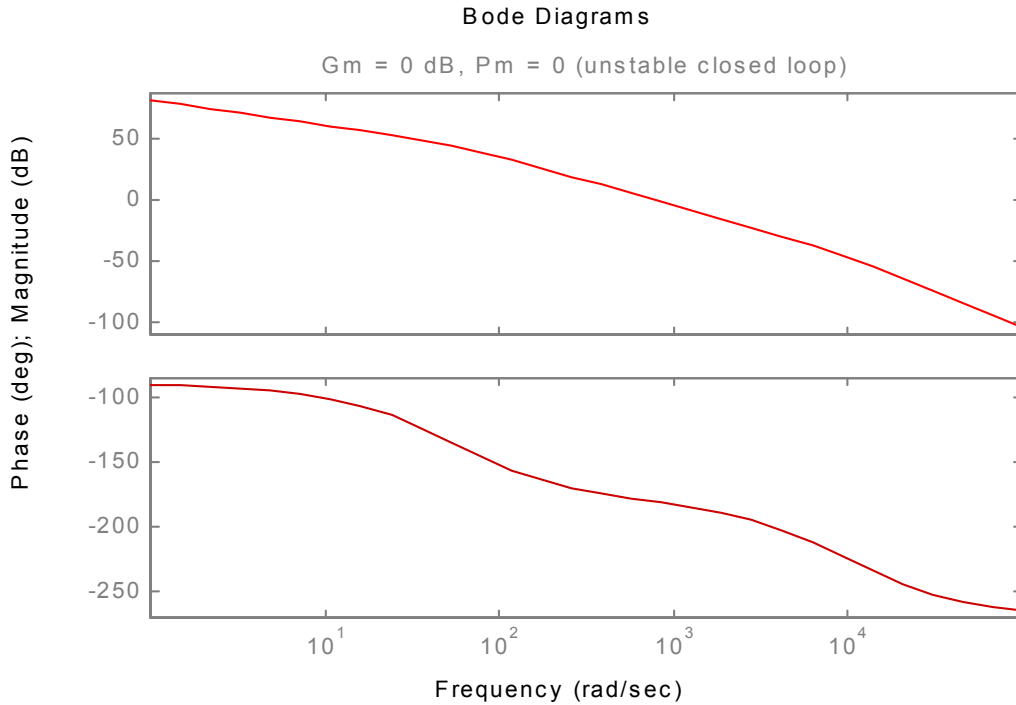


Fig. 9 Respuesta de frecuencia del sistema sin compensar a lazo abierto

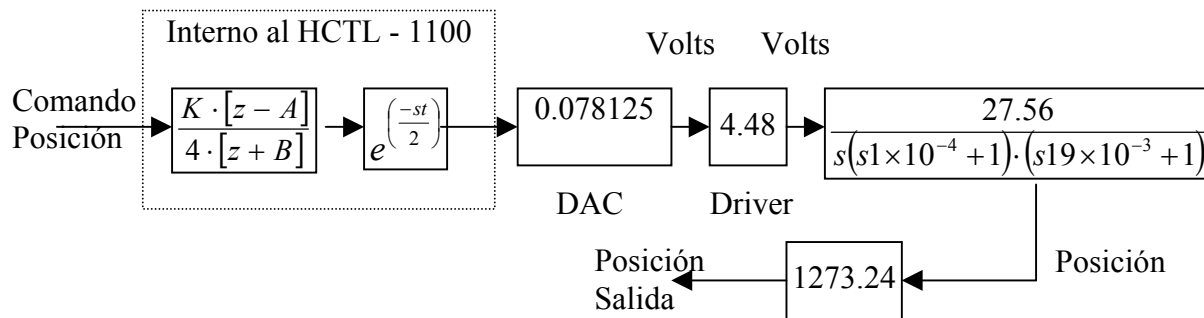


Fig. 10 Sistema con el filtro HCTL 1100

Y la magnitud y fase del sistema a lazo abierto en  $\omega = 377$  rad/seg es:

$$M_{LA}(377) = 20 \cdot \log \left[ \frac{[27.56] \cdot [0.078125] \cdot [4.48] \cdot [(8000 / 2\pi)]}{377 \cdot \sqrt{[1 + ([377] \cdot [0.019])^2]} \cdot \sqrt{[1 + ([377] \cdot [0.0001])^2]}} \right] [db]$$

$$M_{LA}(377) = 13.066 [db]$$

$$P_{LA}(377) = 57.296 \times \left( -\text{tg}^{-1}([377][0.019]) - \text{tg}^{-1}([377][0.0001]) - \left[ \frac{\pi}{2} \right] - \left[ \frac{[377][0.000064]}{2} \right] \right) [grados]$$

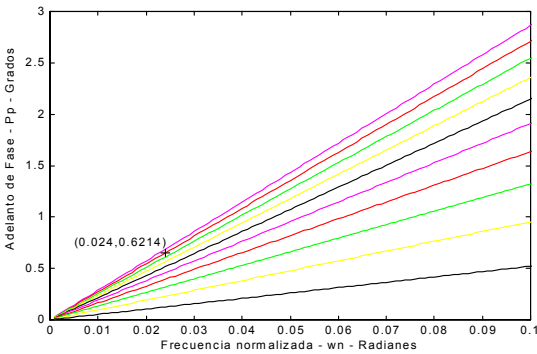
$$P_{LA}(377) = -174.9^\circ$$

$$\Rightarrow MF = 180^\circ + P_{LA}(377) = 180^\circ - 174.9^\circ = 5.096^\circ$$

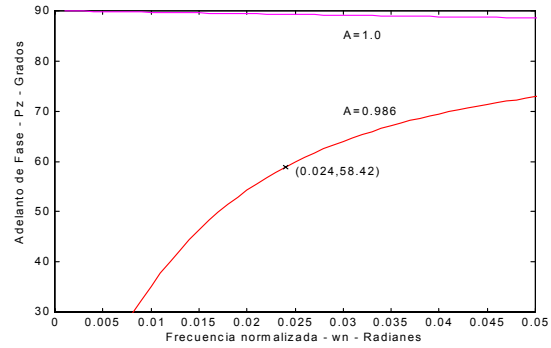
Entrando a la gráfica de la Figura 11 con  $\omega_{NC}' = 0.024128$  y tomando la curva de  $B = 0.9$ , se observa que contribuye con un adelanto de  $0.6214^\circ$ . Se ve que a pesar de ser  $B$  grande la contribución no es significativa.

El margen de fase deseado es  $65^\circ$  pero el sistema sin compensar tiene un margen de fase de  $5.96^\circ$  en consecuencia la compensación debe agregar  $65 - 5.96 = 59.04^\circ$ . Como el polo contribuye con  $0.62^\circ$  en adelante, entonces falta  $59.04^\circ - 0.62^\circ = 58.42^\circ$ .

Lo que falta de adelanto de fase, deber ser contribuido por el Cero, por lo tanto en la Figura 12 se busca cuanto debe valer A para  $\omega_{NC} = 0.024$  y  $P_Z = 58.42^\circ$ , se elige  $A = 0.986$ .



**Figura 11.** Contribución a la Fase del Polo



**Figura 12.** Contribución a la Fase del Cero

La ganancia  $K_F$  que necesita el filtro digital para hacer que toda la ganancia del sistema sea igual a 1 para la frecuencia de corte deseada ( $\omega_c'$ ) es definida como:

$$K_F = \frac{1}{M_{LA}(\omega_c')} = \frac{1}{4.5} = 0.22$$

De la misma forma con las gráficas de la contribución de ganancia debida al polo y al cero se saca la contribución de Magnitud. Para  $B=0.9$  es aproximadamente 0.526 y para  $A = 0.986$  es 0.0276.

En consecuencia la ganancia K que debe ser contribuida por el Filtro digital es:

$$K = \frac{K_F}{\left[ M_P(\omega_{NC}') \right] \cdot \left[ M_Z(\omega_{NC}') \right]} = \frac{0.22}{0.5264 \times 0.0276} ; \quad K = 15.14$$

Por lo tanto los valores que deben ser cargados en los registros del HCTL-1100 son:

$$A = [256] \cdot [0.986] = 252.416 \cong 253 \quad B = [256] \cdot [0.9] = 230.4 \cong 231 \quad K = [4] \cdot [15.14] = 60.56 \cong 61$$

Sin embargo a los fines prácticos se ha dejado un pequeño sobreimpulso para que el afinado definitivo se realice manualmente mediante una realimentación tacométrica analógica de velocidad. En la figura 13 se observa el diagrama en Simulink del sistema con taco y dos gráficas de la respuesta al escalón unitario para diferentes ganancias de taco. Se aplicaron los algoritmos de control compensado al banco de ensayos y al eje 2 del control del robot RSA 2. Figura 14. Los resultados dinámicos obtenidos convalidan en la práctica los aspectos teóricos vertidos. En cuanto a la precisión de posicionamiento, medida con un palpador centesimal, se obtuvieron 836 pulsos del encoder para 12 mm. de desplazamiento en la guía lineal, con lo que se obtiene una discriminación de 14,35 milésimas de mm. (= 14,35  $\mu\text{m}$ .) por cada pulso enviado. Los resultados obtenidos son altamente satisfactorios y permitirán controlar el robot experimental obteniendo una plataforma de referencia exacta para cotejar con los futuros resultados a obtener con el control del mismo robot mediante Redes Neuronales, proyecto de investigación en curso llevado a cabo por el mismo grupo.

Referencias: Se nombran las dos más importantes aplicadas al proyecto:

Hewlett-Packard Application Note 102: "Design of the HCTL-1100 Digital Filter Parameters by the Combination Method" Nov, 1999.

Katsuhito Ogata: "Discrete Time Control Systems" y "Solving Contr. Probl. with Matlab" (95/94).

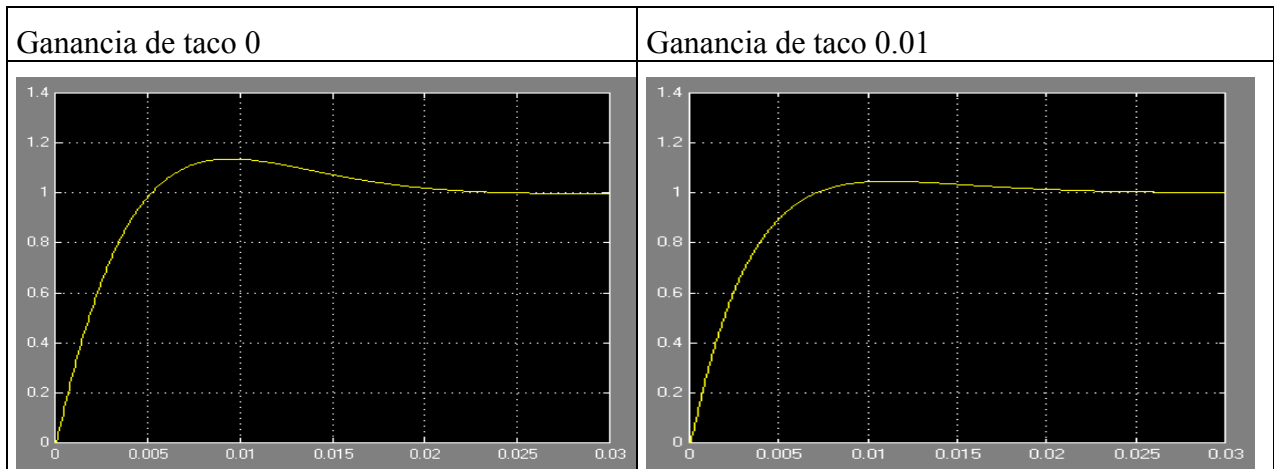
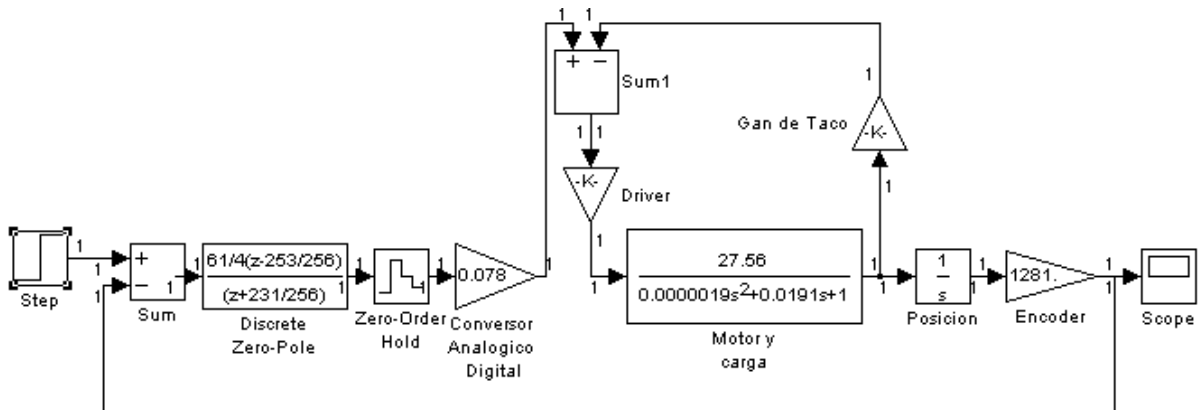


Fig. 13 Simulación y respuestas transitorias con realimentación de tacogenerador

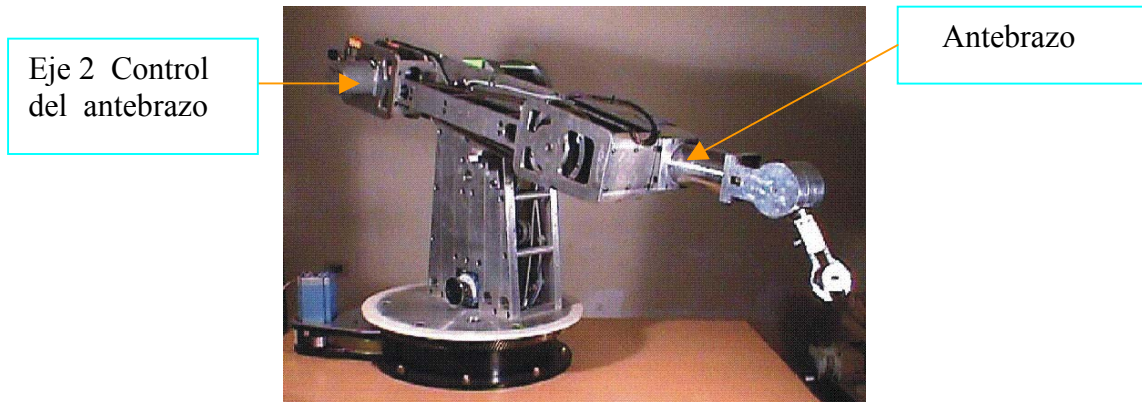


Fig.14 Robot experimental RSA 2