

1ej. 23

UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO

FACULTAD DE INGENIERIA

CONTROL ADAPTABLE DE UN
MOTOR DE CORRIENTE DIRECTA CON
UN MICROPROCESADOR RAPIDO

T E S I S

Que para obtener el título de

p r e s e n t a

Que para obtener el título de
Ingeniero en Computación

p r e s e n t a

GULLERMINA MERINO ZEFERINO

Directores de Tesis

DR. ROMEO ORTEGA MARTINEZ

DR. FRANCISCO GARCIA UGALDE



Ciudad Universitaria, D. F., Octubre, 1987.



Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas Tesis Digitales Restricciones de uso

DERECHOS RESERVADOS © PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis está protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

I N D I C E

INTRODUCCION	1
CAPITULO 1.- ALGORITMO DE CONTROL	3
1.1.- Definición de control adaptable	3
1.2.- Modelo del motor	4
1.3.- Modelo del controlador	8
1.4.- Realización digital	11
CAPITULO 2.- CONTROL DE LA PLANTA SIMULADA	13
2.1.- Implementación del control considerando tres parámetros y una señal cuadrada de referencia	15
2.1.1.- Resultados	18
2.2.- Implementación considerando cinco parámetros y una señal cuadrada de referencia	27
2.2.1.- Resultados	28
2.3.- Implementación considerando cinco parámetros y una señal senoidal de referencia	32
2.3.1.- Resultados	33
CAPITULO 3.- CONTROL DEL MOTOR	41
3.1.- Resultados	50
CONCLUSIONES	60
REFERENCIAS	62
APENDICE A.- Descripción del equipo	63
APENDICE B.- Programas	116
APENDICE C.- Notas técnicas del TMS32010	144

I N T R O D U C C I O N

El control de velocidad en motores de corriente directa es un problema frecuentemente encontrado en aplicaciones industriales. El control de la velocidad no es trivial debido a los efectos de la fricción y a las variaciones de la inercia. En efecto, la fricción es una función no lineal de la velocidad y las variaciones de la inercia no son conocidas. [5].

Tradicionalmente el control de velocidad en motores de C.D. se realiza mediante un sistema de retroalimentación lineal de velocidad y eventualmente de posición [1]. Esta estrategia de control requiere, para su buen funcionamiento, un modelo exacto del motor. Esto significa que los efectos de la fricción y cambios de inercia deben ser conocidos a priori. Este no es siempre el caso en aplicaciones reales.

Un sistema de control adaptable puede ser planteado como posible solución al problema, ya que estos sistemas tienen como vocación primordial el controlar procesos físicos cuyos modelos matemáticos contienen parámetros desconocidos y variantes en el tiempo. [6], [7]. Dicho sistema estima en línea los parámetros del modelo matemático de la planta compensando las variaciones o incertidumbres de la misma con respecto a un modelo de referencia hasta que los parámetros estimados converjan eventualmente a su valor verdadero. [1].

El propósito principal del presente trabajo es el de implementar un algoritmo de control de velocidad para motores de corriente directa sin requerir del conocimiento a priori de los cambios de inercia ni los valores numéricos de la función no lineal que relaciona la velocidad angular con el torque de fricción.

Debido a los últimos adelantos en el desarrollo tecnológico del área -

de semiconductores, los sistemas electrónicos son cada día más versátiles - en cuanto a rapidez y capacidad de cómputo. Así mismo este desarrollo ha - producido una disminución de los costos de fabricación. Esto ha tenido como efecto la proliferación de su uso en el área de control en los últimos años. Tal es el caso del procesador TMS32010 el cual es usado extensivamente en el área de comunicaciones.

Teniendo en cuenta la alta rapidez de cómputo y su relativo bajo costo, la implementación del algoritmo de control será realizado utilizando un sistema digital basado en el procesador TMS32010.

El trabajo realizado durante la preparación de la presente tesis estuvo compuesto de las siguientes etapas:

1. Diseño y construcción de un simulador analógico con el fin de obtener un dispositivo para simular un modelo lineal del motor de C.D. y así poder llevar a cabo los primeros experimentos.

2. Estudio e implementación del algoritmo de control en el microprocesador TMS32010.

3. Diseño y realización de experimentos sobre el simulador.

4. Diseño y realización de experimentos sobre el motor de C.D. del sistema servomodular MS150.

5. Obtención e interpretación de resultados.

Los experimentos realizados al controlar el modelo lineal simulado permitieron validar el algoritmo de control.

Finalmente se procedió a controlar el motor de corriente directa. Los resultados experimentales obtenidos no fueron, sin embargo, los esperados. Una serie de posibles causas son mencionadas justificando los resultados obtenidos para tomarlos en cuenta en un eventual trabajo futuro.

CAPITULO 1.- ALGORITMO DE CONTROL

1.1.- DEFINICION DE CONTROL ADAPTABLE

1.2.- MODELO DEL MOTOR

1.3.- MODELO DEL CONTROLADOR

1.4.- REALIZACION DIGITAL

1.1.- DEFINICION DE CONTROL ADAPTABLE

El diseño de cualquier sistema de control requiere sin excepción del conocimiento de un modelo matemático del proceso o planta a controlar. El modelo dinámico de un motor de C.D. presenta características no lineales y sus parámetros son funciones de variables externas como la inercia.

El control lineal de este tipo de procesos puede ser insatisfactorio si éste trabaja en una amplia zona de operación. [8].

Para controlar la velocidad de un motor de corriente directa puede utilizarse un sistema de control adaptable, con la finalidad de compensar las no linealidades del par de fricción (las cuales son desconocidas) así como las variaciones del momento de inercia.

Un sistema de control adaptable es un sistema, generalmente no lineal, que automáticamente y en línea estima las características dinámicas (como los parámetros del modelo) de la planta, las compara con las características deseadas y usa la diferencia para variar parámetros ajustables del sistema (coeficientes del controlador). [1].

En el diagrama de bloques de la fig. 1.1 se indican las principales partes de un sistema adaptable. Estas son:

1. Identificación de las características dinámicas de la planta.
2. Toma de decisión basada en la identificación de la planta.
3. Modificación o acción basada en la decisión tomada.

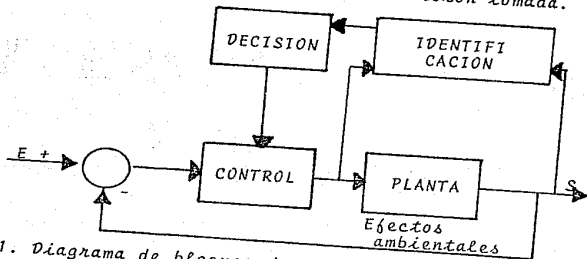


Fig. 1.1. Diagrama de bloques de un sistema de control adaptable.

1.2.- MODELO DEL MOTOR

La velocidad de un motor de corriente directa puede ser controlada en el inducido manteniendo el campo fijo, o bien, controlada en el campo con la corriente de armadura fija.

Para el motor del sistema servomodular MS150 (descrito en el apéndice A), en la conexión por armadura, la armadura se conecta al emisor del circuito con un campo conectado en cada colector de los transistores. [4]. Véase la fig. 1.2.

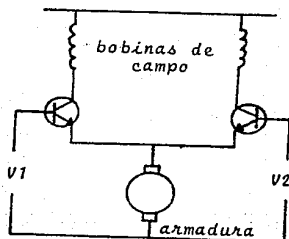


Fig. 1.2. Conexión por armadura

En este caso, al aparecer una fuerza contraelectromotriz (ϵ_{cem}) entre el emisor y tierra, se requiere de un voltaje $V1$ ó $V2$ mínimo para que el motor empiece a girar. A partir de dicho voltaje, éste debe incrementarse para aumentar la velocidad. Esto se muestra gráficamente en la fig. 1.3. - Si cargamos al motor, manteniendo $V1$ ó $V2$ constante, la velocidad disminuye y aumenta la corriente. Ver fig. 1.4.

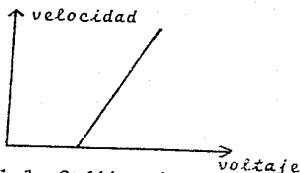


Fig. 1.3. Gráfica de velocidad vs. V_{in}

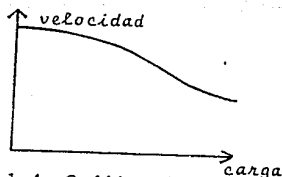


Fig. 1.4. Gráfica de velocidad vs. carga

En la conexión por campo, fig. 1.5., la armadura se conecta al colector de los transistores; con un voltaje muy pequeño se llega a un valor alto de velocidad, lo que representa una ventaja pues tiene una ganancia mayor, sin embargo, tiene características no muy convenientes en sistemas de lazo cerrado en relación con la estabilidad. Si el motor tiene una carga, la velocidad disminuye en forma repentina. Véase fig. 1.6.

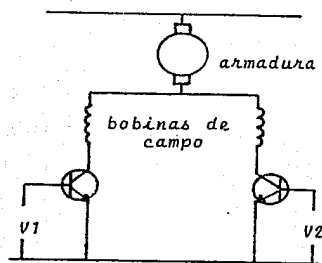


Fig. 1.5. Conexión por campo

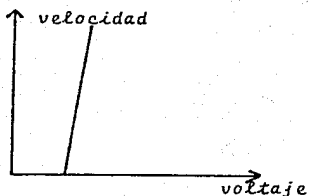
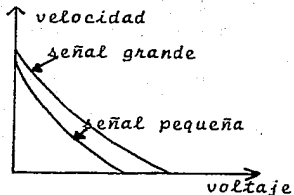


Fig. 1.6. Gráfica de velocidad vs. V_{in}



Gráfica de velocidad vs. carga.

De lo anterior, concluimos que es conveniente que el control se efectúe en el inducido ya que aún cuando se requiere manejar una señal de entrada (V_{in}) mayor, la velocidad se controla más fácilmente.

Un diagrama simbólico del motor controlado por inducido se presenta en la fig. 1.7. [1].

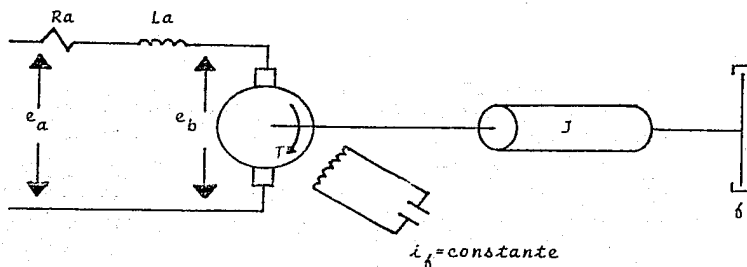


Fig. 1.7. Diagrama del motor controlado por inducido

donde:

R_a = Resistencia del devanado del inducido

L_a = Inductancia del devanado del inducido (despreciable)

i_a = Corriente del devanado del inducido

i_f = Corriente de campo

e_a = Tensión aplicada a la armadura

e_b = Fuerza contraelectromotriz

ω = Velocidad angular

T = Par desarrollado por el motor

J = Momento de inercia equivalente, del motor y carga, con referencia al eje del motor

b = Coeficiente de fricción viscosa equivalente, del motor y carga - referido al eje del motor.

El par T desarrollado por el motor es proporcional al producto de la corriente de inducido i_a y el flujo ϕ del entrehierro, o sea, $T = K_1 \phi i_a$; ϕ , es a su vez proporcional a la corriente de campo por lo que si i_f se mantiene constante, el par se hace directamente proporcional a la corriente de inducido, así,

$$T = K i_a$$

y como,

$$J\dot{\omega} + f(\omega) = T$$

la ecuación diferencial que relaciona el sistema eléctrico y el sistema mecánico del motor es,

$$J\dot{\omega} + f(\omega) = Ki_a$$

1.3. - MODELO DEL CONTROLADO

El modelo matemático que gobierna el comportamiento de un motor de C.D. está dado por la siguiente ecuación diferencial no lineal:

$$J\dot{\omega} + f(\omega) = Ki_a$$

donde ω es la velocidad angular del motor

J es el momento de inercia

i_a es la corriente de armadura

K es una constante

Tradicionalmente, la función $f(\omega)$, esto es, el par de fricción, la cual depende de la velocidad (ω) se modela como:

$$f(\omega) = \begin{cases} a\omega + c & \text{si } \omega > 0 \\ b\omega - d & \text{si } \omega < 0 \end{cases}$$

véase fig. 1.8

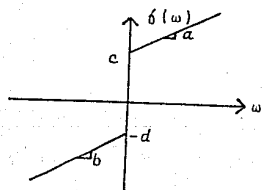


Fig. 1.8. Gráfica del par de fricción

Las características a , b , c y d dependen de la mecánica del motor y en general varían de un motor a otro. Estas características no son proporcionales por el constructor y son, por consecuencia, desconocidas a priori.

En un trabajo reciente [5], el siguiente controlador fue propuesto:

$$i_a = 1/K \{ \hat{\theta}_1 \dot{\omega}_m + \hat{\theta}_2 \omega_m i_1 + \hat{\theta}_3 \omega_m i_2 + \hat{\theta}_4 i_1 + \hat{\theta}_5 i_2 \} \quad (1.1)$$

Su implementación se observa a continuación en el diagrama de bloques - la fig. 1.9.

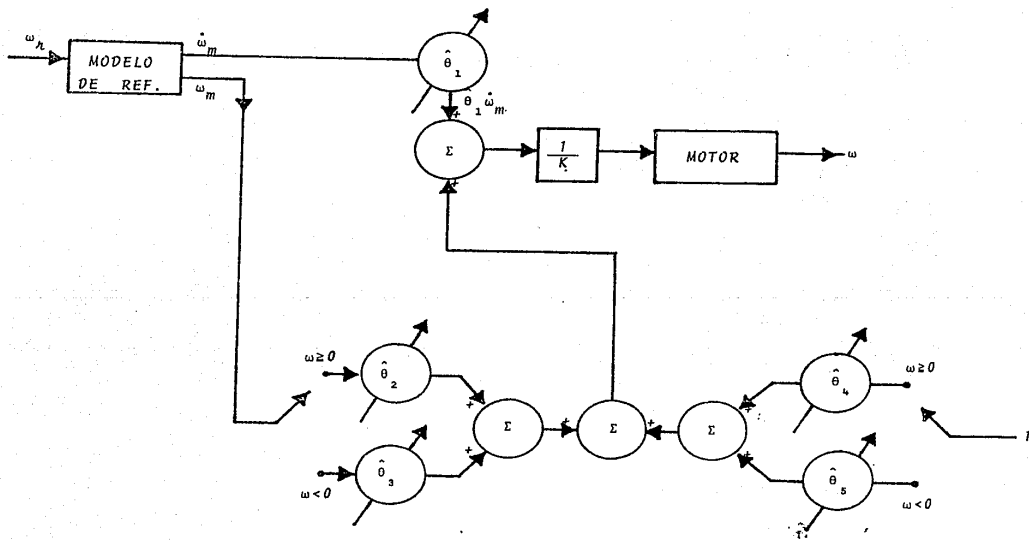


Fig. 1.9. Diagrama de bloques de la implementación del controlador

El objetivo del controlador es determinar una corriente i_a de entrada de tal manera que la velocidad del motor, ω , se aproxime al modelo de referencia, ω_m , ésto es, $e \rightarrow 0$ donde $e \triangleq (\omega - \omega_m)$ y,

$$\omega_m = \frac{1}{\tau_m s + 1} \omega_r \quad (1.2)$$

siendo ω_r la velocidad angular de referencia

El vector de parámetros ajustables del sistema,

$$\hat{\theta} = [\hat{\theta}_1, \hat{\theta}_2, \hat{\theta}_3, \hat{\theta}_4, \hat{\theta}_5]^T = [\hat{J}, \hat{a}, \hat{b}, \hat{c}, -\hat{d}]^T$$

se modifica continuamente hasta cubrir el objetivo del control; el ajuste se realiza mediante la ecuación,

$$\dot{\hat{\theta}} = -\gamma \phi (\omega - \omega_m) \quad (1.3)$$

donde,

γ es un real positivo

$$\phi = [\dot{\omega}_m, i_1 \omega_m, i_2 \omega_m, i_1, i_2]^T$$

i_1 e i_2 serán 1 ó 0 de acuerdo al valor de ω , es decir,

$$i_1 = \begin{cases} 1 & \text{si } \omega \geq 0 \\ 0 & \text{si } \omega < 0 \end{cases} ; \quad i_2 = \begin{cases} 0 & \text{si } \omega \geq 0 \\ 1 & \text{si } \omega < 0 \end{cases}$$

El valor que tome γ depende de la rapidez con que se requiera que el error $(\omega - \omega_m)$ llegue a ser aproximadamente cero. Para valores de γ mayores se convergerá más rápidamente a esta condición.

Sustituyendo ϕ en (1.3) se tiene que,

$$\begin{bmatrix} \dot{\hat{\theta}}_1 \\ \dot{\hat{\theta}}_2 \\ \dot{\hat{\theta}}_3 \\ \dot{\hat{\theta}}_4 \\ \dot{\hat{\theta}}_5 \end{bmatrix} = -\gamma \begin{bmatrix} \dot{\omega}_m \\ i_1 \omega_m \\ i_2 \omega_m \\ i_1 \\ i_2 \end{bmatrix} (\omega - \omega_m) \quad (1.4)$$

El análisis de estabilidad del sistema adaptable presentado ha sido desarrollado en [5].

1.4.- REALIZACION DIGITAL

La realización digital del control se efectúa mediante la transformación rectangular hacia atrás,

$$s \sim \frac{z-1}{Tz}$$

donde T es el periodo de muestreo

Discretizando la ecuación (1.2) se obtiene,

$$\omega_m = \frac{1}{\tau_m \frac{(z-1)}{Tz} + 1} \omega_\lambda$$

$$\omega_m = \frac{Tz}{z(\tau_m + 1) - \tau_m} \omega_\lambda$$

La ecuación de recurrencia en el tiempo es entonces,

$$\omega_m[k] = \frac{1}{(\tau_m + 1)} [T\omega_\lambda[k] + \tau_m \omega_m[k-1]] \quad (1.5)$$

La derivada de $\omega_m[k]$ es aproximadamente, por el método de Euler, igual a,

$$\dot{\omega}_m[k] = \frac{(\omega_m[k] - \omega_m[k-1])}{T} \quad (1.6)$$

En cuanto a la discretización de (1.4)

$$\hat{\theta}[k] = \frac{\hat{\theta}[k] - \hat{\theta}[k-1]}{T}$$

por lo que cada término del vector es,

$$\frac{\hat{\theta}[k] - \hat{\theta}[k-1]}{T} = -\gamma\phi[k][\omega[k] - \omega_m[k]]$$

despejando $\hat{\theta}[k]$

$$\hat{\theta}[k] = \hat{\theta}[k-1] - \gamma T\phi[k][\omega[k] - \omega_m[k]] \quad (1.7)$$

Por tanto,

$$i_\alpha[k] = 1/K [\hat{\theta}_1[k]\dot{\omega}_m[k] + \hat{\theta}_2[k]\omega_m[k]i_1 + \hat{\theta}_3[k]\omega_m[k]i_2 + \hat{\theta}_4[k]i_1 + \hat{\theta}_5[k]i_2] \quad (1.8)$$

Las ecuaciones (1.5), (1.6) (1.7) y (1.8) fueron implantadas en el TMS32010. La fig. 1.10 muestra la configuración del sistema de control.

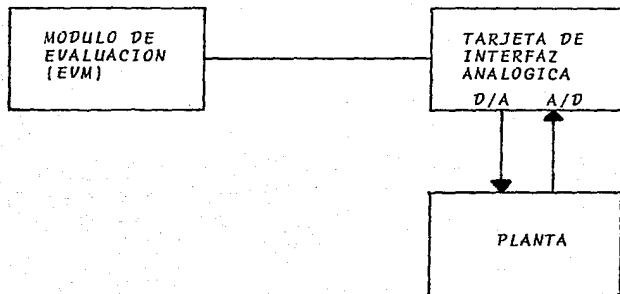


Fig. 1.10. Configuración del sistema de control

En el módulo de evaluación del TMS32010 (EVM), se compara la salida de la planta (simulador de F.T. o sistema servomodular MS150) con la señal de referencia deseada ($\omega - \omega_m$). Basándose en esta comparación, se modifican los parámetros ajustables (θ) y se calcula la señal de control i_a .

La tarjeta de interfaz analógica nos permite la comunicación entre el EVM y la planta. (La información referente al EVM y la tarjeta de interfaz se encuentra en el apéndice A).

En los próximos dos capítulos se presentan los resultados experimentales. Primeramente al controlar un modelo lineal analógico del motor y en se guida el motor real.

CAPITULO 2.- CONTROL DE LA PLANTA SIMULADA

- 2.1.- IMPLEMENTACION DEL CONTROL CONSIDERANDO TRES PARAMETROS Y UNA SENAL CUADRADA DE REFERENCIA**
 - 2.1.1.- RESULTADOS**
- 2.2.- IMPLEMENTACION CONSIDERANDO CINCO PARAMETROS Y UNA SENAL CUADRADA DE REFERENCIA**
 - 2.2.1.- RESULTADOS**
- 2.3.- IMPLEMENTACION CONSIDERANDO CINCO PARAMETROS Y UNA SENAL SENOIDAL DE REFERENCIA**
 - 2.3.1.- RESULTADOS**

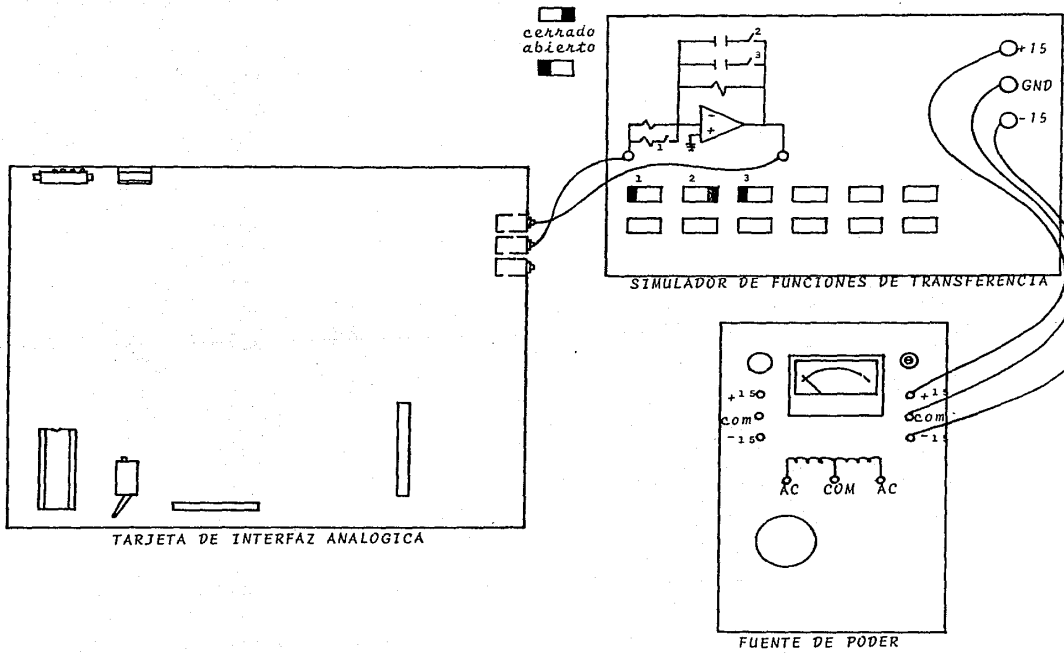
En el presente capítulo se presentarán los resultados experimentales -
tenidos al controlar un modelo lineal del motor de corriente directa simu-
lado analógicamente.

El objetivo de esta serie de experimentos fue el de verificar el buen-
funcionamiento del sistema de control en un ambiente de parámetros conoci-
dos. Esta primera etapa experimental permitió validar el algoritmo de con-
trol ya que los resultados experimentales concordaron con los esperados.

Lo que permite determinar si el control se efectúa adecuadamente, es -
el hecho de que ω tienda a ser igual a ω_m y los parámetros ajustables tomen
sus valores verdaderos ($\hat{\theta} = \theta$ cuando $\omega = \omega_m$), después de un tiempo determinado.

Utilizando el simulador de funciones de transferencia se verifica el -
buen funcionamiento del controlador ya que el vector θ es conocido.

El diagrama de conexiones entre la interfaz y el simulador se muestra
en la fig. 2.1.



cerrado
 abierto

Fig. 2.1 Diagrama de conexiones entre la interfaz y el simulador de F.T.

Para simular el comportamiento del motor (ec. diferencial de primer orden, $J\dot{\omega} + f(\omega) = K i_a$), se considera $f(\omega)$ constante, obteniéndose la F.T.,

$$\frac{\omega}{i_a} = \frac{K}{Js + f} \quad (2.1)$$

La gráfica del par de fricción en este caso es, (ver fig. 2.2)

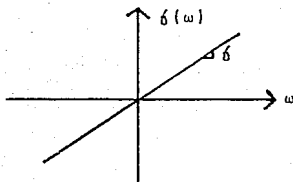


Fig. 2.2 Gráfica del par de fricción $f(\omega)$

Comparando con la fig. 1.8 y como $\hat{\theta}_1 = J$; $\hat{\theta}_2 = a$; $\hat{\theta}_3 = b$; $\hat{\theta}_4 = c$; $\hat{\theta}_5 = -d$, se tiene que los parámetros $\hat{\theta}_2$ y $\hat{\theta}_3$ son iguales a f y $\hat{\theta}_4$ y $\hat{\theta}_5$ iguales a cero.

Tomando en cuenta que estos dos últimos parámetros son cero, se pueden excluir del modelo del controlador, o bien incluirlos para comprobar que efectivamente toman el valor de cero cuando $\omega = \omega_m$. Por este motivo se analizarán los dos casos.

2.1. IMPLEMENTACION DEL CONTROLADOR CONSIDERANDO TRES PARAMETROS Y UNA SENAL CUADRADA DE REFERENCIA.

Del simulador se obtiene la función de transferencia,

$$H(S) = \frac{1}{S + 1} \quad (2.2)$$

comparando con (2.1) $K=1$, $J=1$ y $f=1$ por lo que el vector θ es,

$$\theta = [J, f, f, 0, 0]^T = [1, 1, 1, 0, 0]^T$$

En lo que respecta al controlador, se selecciona un valor de $\tau_m = 1$; entonces el modelo de referencia será,

$$\omega_m = \frac{1}{S + 1} \omega_r \quad (2.3)$$

donde ω_r es una señal cuadrada de frecuencia 0.05Hz. (que se obtiene con un

valor de 50 en CDAT) y 2.5 V_{pp} (4096 y -4096 en el convertidor D/A). Tomando un periodo $T=0.2$ seg., las ecuaciones de recurrencia son,

$$\omega_m(k) = \frac{\omega_m(k-1)}{1.2} + \frac{0.2}{1.2} \omega_n(k) \quad (2.4)$$

$$\dot{\omega}_m(k) = \frac{(\omega_m(k) - \omega_m(k-1))}{0.2} \quad (2.5)$$

$$\hat{\theta}_1(k) = \hat{\theta}_1(k-1) - 0.2\gamma\dot{\omega}_m[\omega(k) - \omega_m(k)] \quad (2.6)$$

$$\hat{\theta}_2(k) = \hat{\theta}_2(k-1) - 0.2\gamma\omega_m[\omega(k) - \omega_m(k)]i_1 \quad (2.7)$$

$$\hat{\theta}_3(k) = \hat{\theta}_3(k-1) - 0.2\gamma\omega_m[\omega(k) - \omega_m(k)]i_2 \quad (2.8)$$

El programa que calcula i_a es un programa en ensamblador. Consta de un programa principal, una subrutina de cálculo y una subrutina de retardo. (Ver apéndice B prog. 1).

En el programa principal se genera la velocidad de referencia ω_n . El periodo de esta señal se obtiene mediante el producto del tiempo especificado en la subrutina de retardo y dos veces el número de datos que indica el contador CDAT.

En la subrutina de cálculo, se lee la muestra de la velocidad ω a la salida de la planta, se calculan los parámetros $\hat{\theta}$ y la corriente i_a .

La subrutina de retardo se utiliza para simular ω_m , reteniendo cada valor de $\omega_m(k)$, un periodo de muestreo T . La siguiente tabla indica el valor que deberán tener los contadores CONTA y CONTD para un tiempo de retardo determinado.

CONTA	CONTD	TIEMPO DE RETARDO
6	7FFF	300 mS.
5	7FFF	240 mS.
3	7FFF	150 mS.
2	7FFF	100 mS.
3	1FFF	40 mS.

CONTA	CONTD	TIEMPO DE RETARDO
1	1FFF	12 mS.
1	FFF	7 mS.
1	7FF	3.2 mS.
1	3FF	1.6 mS.
1	1FF	0.8 mS.
1	FF	0.4 mS.

Las variables que intervienen en el programa se especifican en el apéndice B.

Debido a que la mayor parte de las variables toman valores menores que la unidad es necesario utilizar un formato, en este caso, el formato Q[12]. Así, los datos: constante K, periodo de muestreo T, gama, amplitud de ω_n , ENT, alfa y beta deberán estar multiplicados por 4096 o 2^{12} . Otros datos como CDAT, que es el contador del que depende la frecuencia de la señal ω_n , así como TINV que es el inverso del periodo de muestreo, no requieren de un formato.

Para observar que el controlador cumple con su objetivo, se grafican la variación de los parámetros ajustables (vector $\hat{\theta}$) y la salida de la planta w .

La tarjeta de interfaz analógica solo cuenta con una salida y una entrada que se utilizan, como salida de la señal de control i_a y entrada de la muestra de w , respectivamente. Por ello, no tenemos acceso a los valores del vector $\hat{\theta}$ durante la ejecución del programa para ser graficados como sucede con la salida de la planta y es necesario almacenarlos en la memoria de expansión.

Se almacena el vector $\hat{\theta}$ durante los semiciclos positivos de la señal para determinado número de ciclos, al cabo de los cuales se detiene el programa.

Para detener el programa después del número de ciclos, indicado por -

el contador CONS, se utilizan las instrucciones:

	BNZ	L2
	LACK	CT
	TBLR	CDAT
	LAC	CONS
	SUB	DELTA
	AND	MASK3
	SACL	CONS
	BNZ	L1
	B	L11
SUB3	MPY	ENT
	PAC	
	SACH	SUM2,4

* L11 Breakpoint

Una vez en la memoria de expansión, con ayuda del programa 4 (apéndice B), se grafican dichos parámetros.

La palabra de control, tanto para el programa 1 (algoritmo de control) como para el programa 4 es 58. Se carga en el registro de control de la tarjeta con una instrucción OUT por el puerto 0. La dirección a partir de la cual deseamos almacenar los parámetros en la memoria de expansión se carga también con una instrucción OUT, por el puerto 4.

Se debe tomar en cuenta que estas instrucciones solo se incluirán en el programa 1 cuando se grafiquen los parámetros δ .

Para almacenar cada dato, se utiliza la instrucción:

OUT <DMA>,5

Se coloca después de que éste es guardado en la localidad de memoria de datos respectiva, ya sea, TETA1, TETA2 ó TETA3.

2.1.1. Resultados

Al iniciar los experimentos, se utilizó un periodo de muestreo $T=0.2$ -seg. en la realización digital del controlador (ecs. 1.5, 1.6, 1.7 y 1.8).

Para dicho valor de T , no se obtiene el resultado deseado ya que, de acuerdo al objetivo del controlador, la señal de salida, ω , debe ir aproximándose cada vez más a la señal de referencia, ω_m ; por ejemplo, para $\gamma=0.1$

$\gamma=7$, como lo muestra la fig. 2.3, esto no ocurre; en los dos primeros ciclos se observan cambios lo que indicaría que ω tiende a ω_m , sin embargo, - después de varios ciclos más, ω no mejora, es decir, no tiende a ω_m .

En pruebas con otros valores de T , (fig. 2.4), para un valor de $T=0.04$ seg. sucede lo mismo que en el caso anterior aunque ω se acerca mucho más a ω_m .

Al disminuir todavía más el periodo de muestreo, para $T=0.012$ seg. y $T=0.007$ seg. (figs. 2.5 y 2.6), respectivamente, después de 3 ciclos para la señal con $\gamma=7$ y 10 ciclos para la señal con $\gamma=0.1$, consideramos que - - = ω_m pues solo guardan una diferencia en amplitud de 2% aproximadamente.

De aquí podemos concluir que el controlador se comporta correctamente ara periodos de muestreo pequeños y que ω converge más rápidamente a ω_m - ara valores de gama mayores.

En lo que respecta al vector de parámetros $\hat{\theta}$, al inicio del capítulo, se menciona que el vector θ , en este caso, es conocido. En efecto, para la función de transferencia de la ecuación (2.2) se tiene que el vector θ es,

$$\theta = [1, 1, 1, 0, 0]^T$$

Así mismo, también se mencionó que podemos incluir o no los parámetros $\hat{\theta}_4$ y $\hat{\theta}_5$. En este caso no se incluyen. De esta manera, al aproximarse a ω_m , los parámetros $\hat{\theta}_1$, $\hat{\theta}_2$ y $\hat{\theta}_3$ deberán tender a θ_1 , θ_2 y θ_3 , es decir, 1. Debemos aclarar que no necesariamente deben tomar el valor de 1, pero un valor cercano, que dependiendo de gama en la expresión (1.3) se acercarán más rápidamente.

Lo anterior se comprueba analizando las figs. 2.7 y 2.8 ya que para - mismo número de ciclos de la señal, ω , el vector $\hat{\theta}$ para $\gamma=7$ es,

$$\hat{\theta} = [1.25, 1.03, 0.9084]$$

para $\gamma=0.1$ es,

$$\hat{\theta} = [1.32, 0.78, 0.63]$$

Hay que hacer notar que se pueden obtener periodos de muestreo meno -

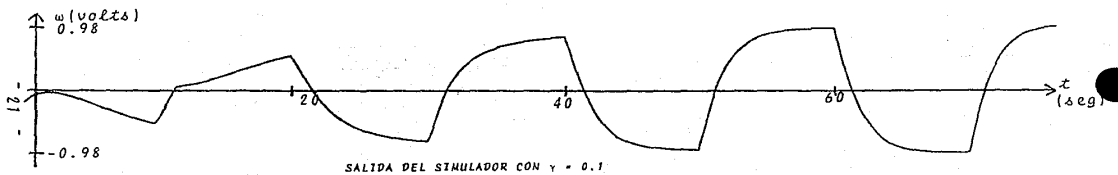
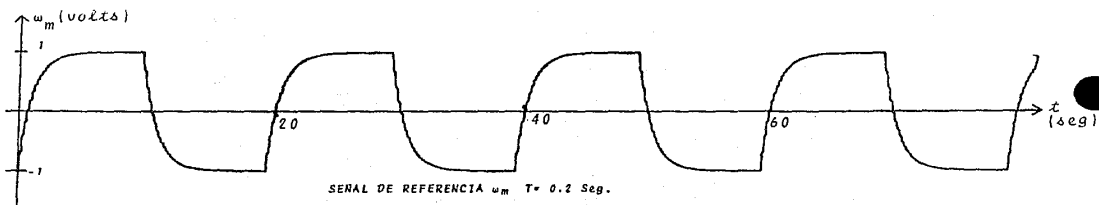
res a los utilizados por tratarse de un microprocesador muy rápido ya que no se tienen problemas con el tiempo de cálculo. Sin embargo, a medida que T se hace pequeño, se pierde exactitud en la generación de ω_m .

Los datos que se modifican en el programa para cada valor de T son los siguientes. TR, ALPH y BET se encuentran en formato Q[12].

$T = 0.2 \text{ seg.}$	$T = 0.15 \text{ seg.}$	$T = 0.04 \text{ seg.}$
TR = 819	TR = 614	TR = 163
CT = 50	CT = 67	CT = 250
ALPH = 3413	ALPH = 3561	ALPH = 3938
BET = 682	BET = 534	BET = 157
TINV = 5	TINV = 6	TINV = 25

$T = 0.007 \text{ seg.}$
TR = 28
CT = 1428
ALPH = 4067
BET = 28
TINV = 142

El contador CDAT se modifica de tal manera que para los cuatro periodos de muestreo se obtenga una señal de referencia ω_m de frecuencia 0.05 Hz.



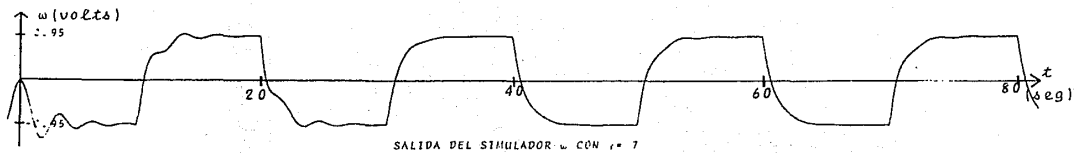
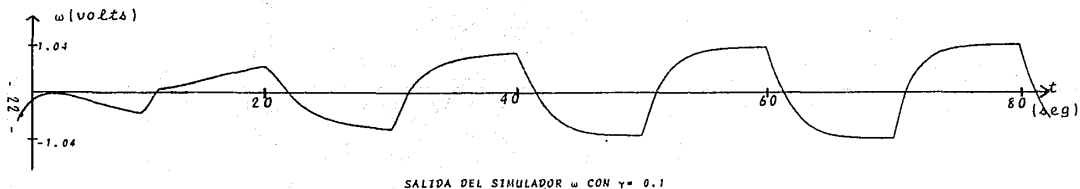
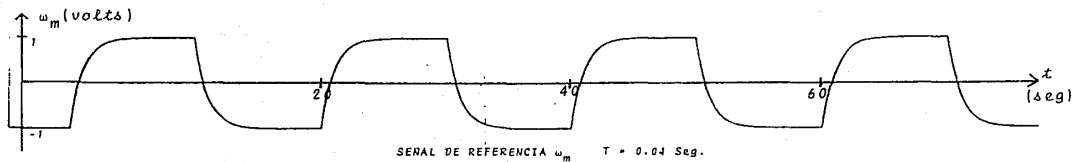
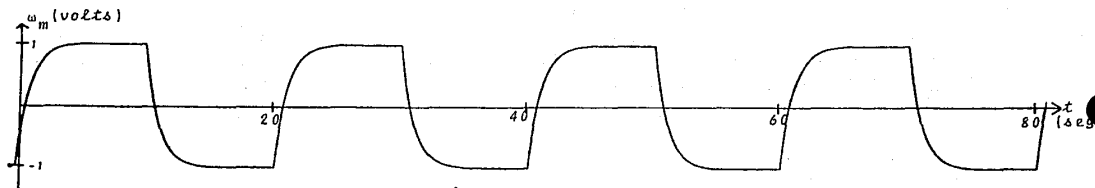
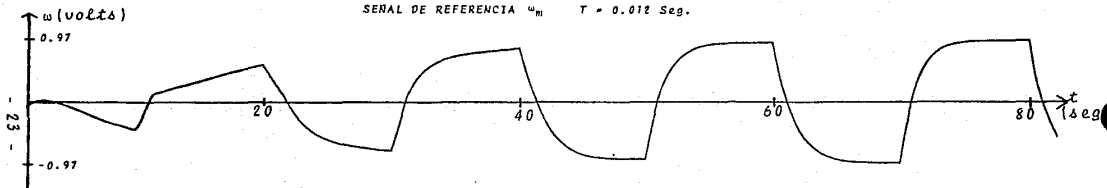


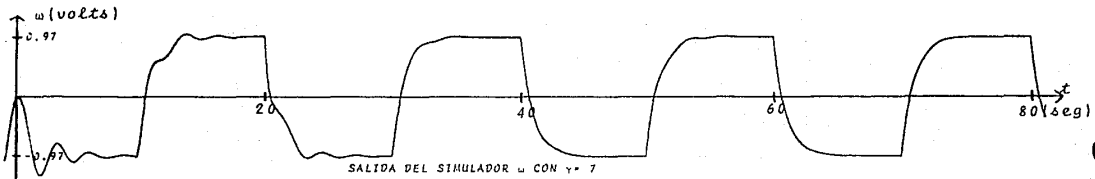
Fig. 2.4.



SERIAL DE REFERENCIA ω_m $T = 0.012$ Seg.

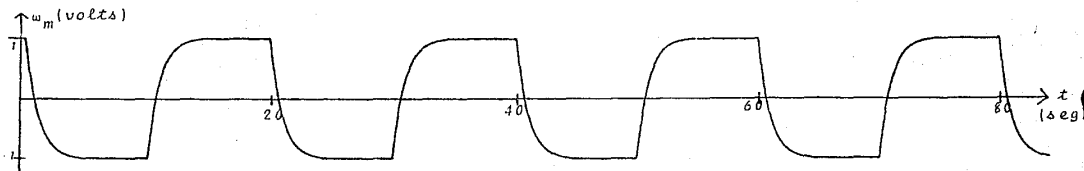


SALIDA DEL SIMULADOR ω CON $\gamma = 0.1$

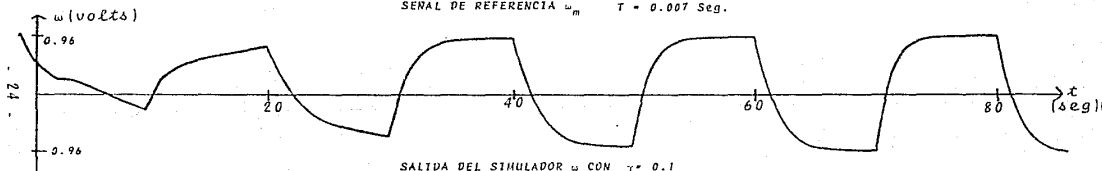


SALIDA DEL SIMULADOR ω CON $\gamma = 7$

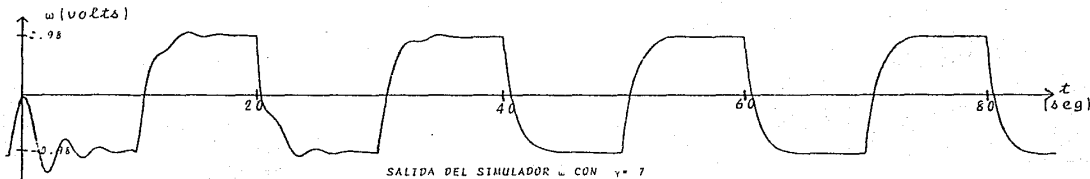
Fig. 2.5.



SEÑAL DE REFERENCIA ω_m $T = 0.007$ Seg.



SALIDA DEL SIMULADOR ω CON $\gamma = 0.1$



SALIDA DEL SIMULADOR ω CON $\gamma = 7$

Fig. 2.6.

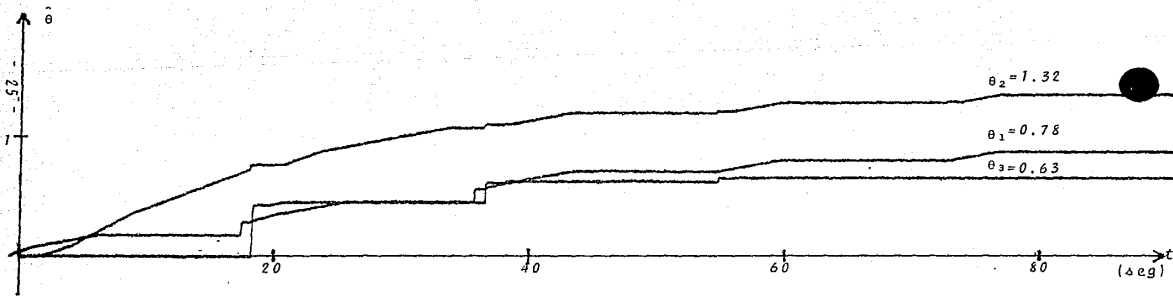
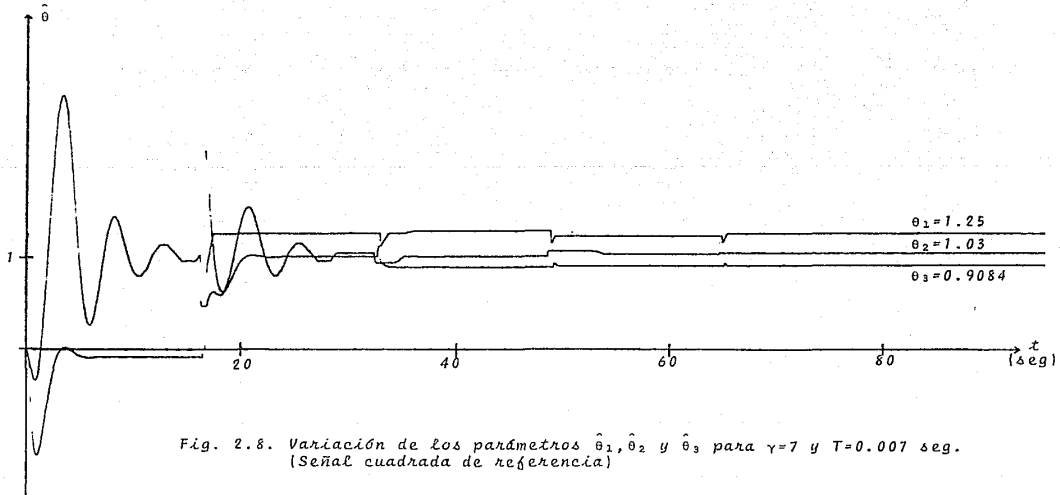


Fig. 2.7. Variación de los parámetros $\hat{\theta}_1$, $\hat{\theta}_2$ y $\hat{\theta}_3$ para $\gamma=0.1$ y $T=0.007$ seg. (Señal cuadrada de referencia).



2.2.- IMPLEMENTACION CONSIDERANDO CINCO PARAMETROS Y UNA SENAL CUADRADA DE REFERENCIA.

La función de transferencia que simula la del motor así como el modelo de referencia ω_m son:

$$H(S) = \frac{1}{S+1} \quad y \quad \omega_m = \frac{1}{S+1} \omega_r \quad (2.9)$$

respectivamente como en el caso anterior.

ω_r es una señal cuadrada de frecuencia 0.05 Hz. y amplitud de 2.5 V_{pp}.

En la implementación con tres parámetros, el periodo de muestreo con el que se obtuvieron los mejores resultados fue el de 0.007 seg. Por esta razón, dicho valor de T en la implementación con cinco parámetros es utilizado.

Las ecuaciones para el cálculo de i_a con T=0.007 seg, son las siguientes:

$$\omega_m(k) = \frac{\omega_m(k-1)}{1.007} + \frac{0.007}{1.007} \omega_r(k) \quad (2.10)$$

$$\dot{\omega}_m(k) = \frac{(\omega_m(k) - \omega_m(k-1))}{1.007} \quad (2.11)$$

$$\hat{\theta}_1(k) = \hat{\theta}_1(k-1) - 0.007\gamma\dot{\omega}_m[\omega(k) - \omega_m(k)] \quad (2.12)$$

$$\hat{\theta}_2(k) = \hat{\theta}_2(k-1) - 0.007\gamma\omega_m[\omega(k) - \omega_m(k)]i_1 \quad (2.13)$$

$$\hat{\theta}_3(k) = \hat{\theta}_3(k-1) - 0.007\gamma\omega_m[\omega(k) - \omega_m(k)]i_2 \quad (2.14)$$

$$\hat{\theta}_4(k) = \hat{\theta}_4(k-1) - 0.007\gamma[\omega(k) - \omega_m(k)]i_1 \quad (2.15)$$

$$\hat{\theta}_5(k) = \hat{\theta}_5(k-1) - 0.007\gamma[\omega(k) - \omega_m(k)]i_2 \quad (2.16)$$

La señal de control i_a se calcula en el programa 2 (apéndice B). Este programa es semejante al prog. 1 (implementación con 3 parámetros) pues se lo se incluye el cálculo de los parámetros $\hat{\theta}_4$ y $\hat{\theta}_5$ así como dos términos -

más en la ecuación del controlador (ec. 1.8).

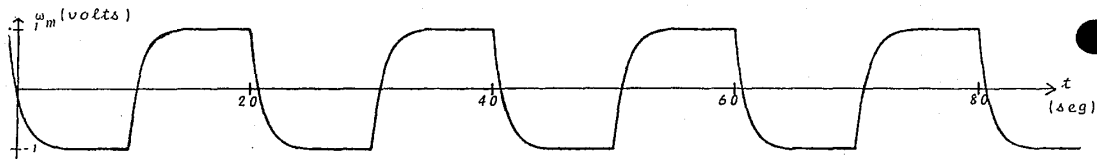
2.2.1. Resultados.

De acuerdo a las pruebas realizadas para varios valores de gama, se observa que el comportamiento del controlador es el correcto: ω se aproxima a ω_m rápidamente para valores de gama mayores. Como ejemplo se graficaron las señales de salida, ω , con $\gamma=0.1$ y $\gamma=7$. (ver figs. 2.9(a) y 2.9(b))

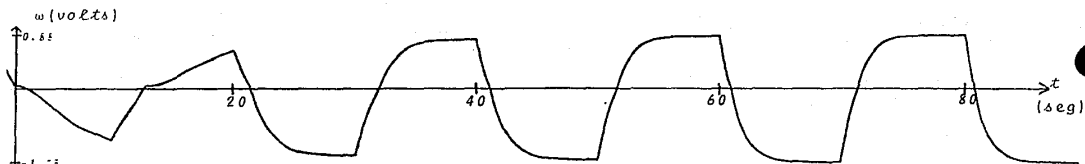
Considerando los cinco parámetros, se espera que el vector $\hat{\theta}$ se aproxime a $\theta = [1, 1, 1, 0, 0]^T$.

Para comprobar lo anterior, se graficó la variación del vector $\hat{\theta}$ considerando un valor de gama igual a 7. (fig. 2.10). Cuando $\omega = \omega_m$ aproximadamente, el vector de parámetros $\hat{\theta}$ es el siguiente:

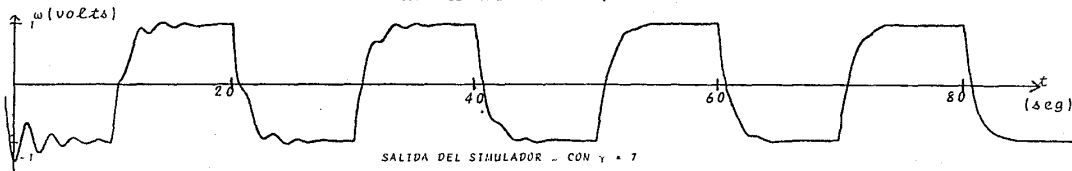
$$\hat{\theta} = [1.24, 1.13, 1.4, -0.09, -0.19]$$



SERIAL DE REFERENCIA ω_m T=0.007 Seg.

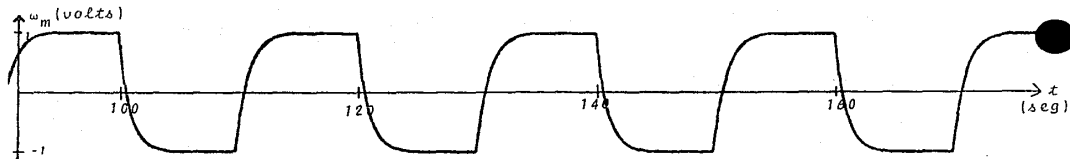


SALIDA DEL SIMULADOR ω CON $\gamma = 0.1$

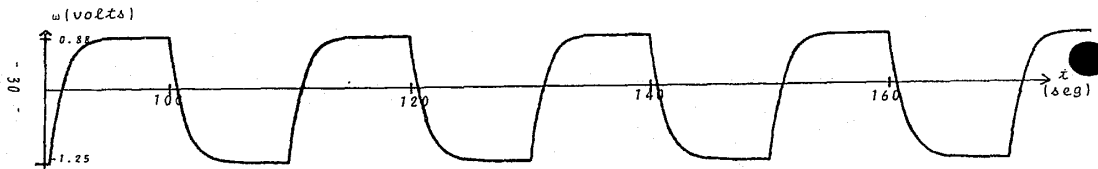


SALIDA DEL SIMULADOR ω CON $\gamma = 7$

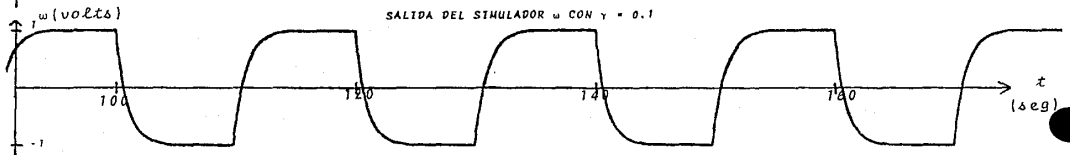
Fig. 2.9. (a)



SERIAL DE REFERENCIA ω_m $T=0.007$ Seg.



SALIDA DEL SIMULADOR ω CON $\gamma = 0.1$



SALIDA DEL SIMULADOR ω CON $\gamma = 7$

Fig. 2.9. (b). (Continúa de 2.9. (a))

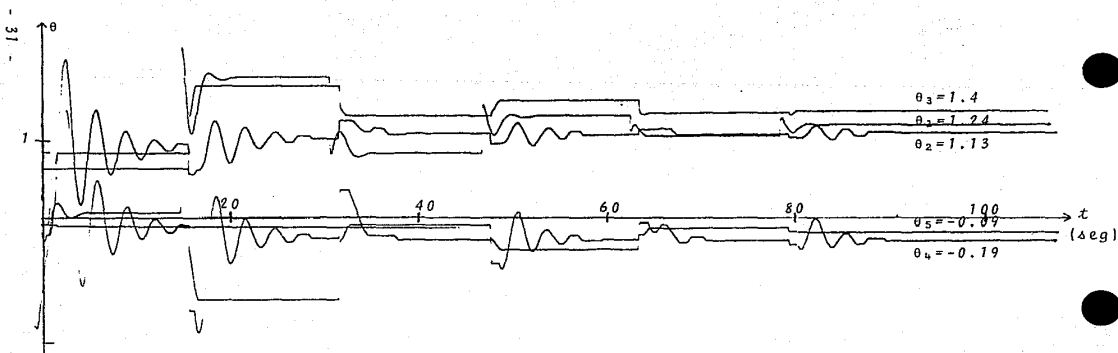


Fig. 210. Variación de los parámetros $\hat{\theta}_1$, $\hat{\theta}_2$, $\hat{\theta}_3$, $\hat{\theta}_4$ y $\hat{\theta}_5$ para $\gamma=7$ y $T=0.007$ seg. [Señal cuadrada de referencia].

2.3.- IMPLEMENTACION CONSIDERANDO CINCO PARAMETROS Y UNA SEÑAL SENOIDAL DE REFERENCIA.

$$H(S) = \frac{1}{S + T} \quad (\text{F.T. simulada}) \quad (2.17)$$

$$\omega_m = \frac{1}{S + T} \omega_r \quad (\text{Modelo de referencia}) \quad (2.18)$$

donde ω_r es una señal senoidal

Para generar las ecuaciones de recurrencia que se utilizan en el cálculo de i_a , se elige un valor de T de tal manera que se nos facilite el muestreo de la señal ω_r , ya que es generada dentro del módulo de evaluación a partir de una tabla de 128 elementos. También debe ser pequeño para que el controlador funcione correctamente. Este período de muestreo es $T = 0.0064$ seg. Dichas ecuaciones son las siguientes.

$$\omega_m(k) = \frac{\omega_m(k-1)}{1.0064} + \frac{0.0064}{1.0064} \omega_r(k) \quad (2.19)$$

$$\dot{\omega}_m(k) = \frac{(\omega_m(k) - \omega_m(k-1))}{1.0064} \quad (2.20)$$

$$\hat{\theta}_1(k) = \hat{\theta}_1(k-1) - 0.0064 \dot{\omega}_m \gamma [\omega(k) - \omega_m(k)] \quad (2.21)$$

$$\hat{\theta}_2(k) = \hat{\theta}_2(k-1) - 0.0064 \dot{\omega}_m \gamma [\omega(k) - \omega_m(k)] i_1 \quad (2.22)$$

$$\hat{\theta}_3(k) = \hat{\theta}_3(k-1) - 0.0064 \dot{\omega}_m \gamma [\omega(k) - \omega_m(k)] i_2 \quad (2.23)$$

$$\hat{\theta}_4(k) = \hat{\theta}_4(k-1) - 0.0064 \gamma [\omega(k) - \omega_m(k)] i_1 \quad (2.24)$$

$$\hat{\theta}_5(k) = \hat{\theta}_5(k-1) - 0.0064 \gamma [\omega(k) - \omega_m(k)] i_2 \quad (2.25)$$

El programa que genera i_a consta de un programa principal, una subrutina de generación de la señal senoidal, una subrutina de cálculo y una subrutina de retardo. (ver apéndice B prog. 3).

En el programa principal se muestrea la señal ω_r . El contador CONT1 indica qué dato de la tabla es el que se toma. (El método utilizado para -

la generación de la señal senoidal se encuentra en las notas técnicas del-TMS32010. Apéndice C).

Dentro de la subrutina de generación de la señal senoidal, el retardo se utiliza para generar bajas frecuencias (menores a 1 Hz). En la siguiente tabla se muestran los valores de CONTA, CONTD y CONTI para varias frecuencias.

CONTA	CONTD	CONTI	FRECUENCIA (Hz)
1	7FF	2	0.0125
1	3FF	4	0.0185
1	1FF	8	0.0384
1	FF	16	0.0769
1	7F	30	0.1388
1	3F	60	0.2777
1	1F	120	0.5
1	F	240	0.8695

En la subrutina de cálculo, se lee la muestra de la velocidad ω a la salida de la planta, se calculan los parámetros $\hat{\theta}$ y la corriente i_a .

2.3.1. Resultados

Se realizaron pruebas con todas las frecuencias de la tabla anterior, así como para varios valores de gama.

Se obtuvo lo esperado pues ω se aproxima más rápidamente a ω_m con valores de gama mayores. Como ejemplo se graficó la salida de la planta, utilizando $\gamma=7$, para señales de referencia de frecuencias 0.0125 Hz., 0.0185-Hz. y 0.0384 Hz. (figs. 2.13, 2.14 y 2.15, respectivamente).

La función de transferencia simulada, es la misma que para la implementación en los dos puntos anteriores (2.1 y 2.2), por tanto el vector θ_{es} :

$$\theta = [1, 1, 1, 0, 0]$$

Utilizando el mismo método que en los puntos 2.1 y 2.2 para la grafi-

cación de los parámetros, la memoria de expansión no fue suficientemente grande para almacenar los datos hasta el momento en que $\omega = \omega_m$. En la fig. 2.11 vemos que $\hat{\theta}_5$ aún no se acerca a 0. Con el fin de considerar la variación de los parámetros durante un mayor número de ciclos, se tomaron los valores del vector $\hat{\theta}$ después de cada ciclo, directamente de las localidades de memoria de datos. El programa se detiene para realizar las lecturas de cada ciclo.

Para detener el programa después de cierto número de ciclos (indicado por el contador CONS) se utilizan las instrucciones:

	LT	BETA
	CALL	SUB3
	LAC	ALFA
	BZ	L10
	B	L1
L10	LAC	CONS
	SUB	DELTA
	AND	MASK
	SACL	CONS
	BNZ	L1
SWAVE1	LAC	ALFA, 8
	SACH	TEMP

* L11 Breakpoint

Estas instrucciones se incluyen en el programa 3 (apéndice B) solo -- cuando se grafican los parámetros.

La siguiente tabla muestra el vector de parámetros $\hat{\theta}$, en formato - - Q[12], para 10 ciclos de la señal ω de frecuencia 0.0185 Hz.

	TETA1	TETA2	TETA3	TETA4	TETA5
1	7087	-990	2253	-315	-2058
2	9144	2968	5740	2569	-973
3	5221	5694	6605	1239	1442
4	2842	5366	5422	-406	1547
5	3107	4153	4453	-784	1148
6	4354	3734	4303	-245	364
7	4777	3798	4280	140	175

	TETA1	TETA2	TETA3	TETA4	TETA5
8	4738	4014	4202	175	182
9	4616	4071	4257	70	196
10	4672	4031	4157	119	147

De acuerdo al formato Q[12], 4096 equivale a la unidad por lo que conclumos, de la tabla anterior, que a partir del sexto ciclo de la señal, los parámetros han llegado a valores aproximados al vector θ .

Estos datos son graficados mediante el prog. 5 (apéndice B). Ver fig. 2.12

Aún cuando no se grafican todos los valores de $\hat{\theta}$, con los 10 datos - que se tienen, se comprueba que los parámetros tienden a sus valores verdaderos (vector θ). El vector $\hat{\theta}$ correspondiente al décimo ciclo es:

$$\hat{\theta} = [1.14, 1.01, 0.98, 0.029, 0.035]$$

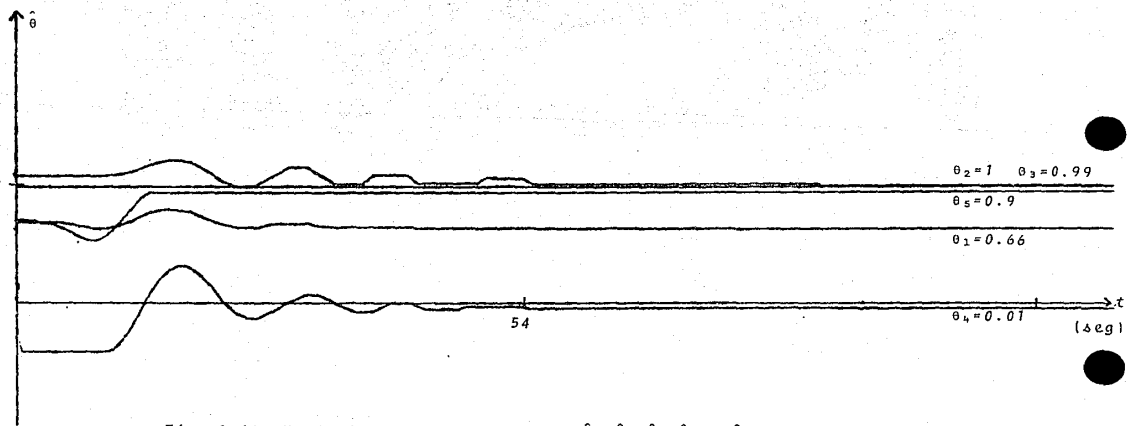


Fig. 2.11. Variación de los parámetros $\hat{\theta}_1, \hat{\theta}_2, \hat{\theta}_3, \hat{\theta}_4$ y $\hat{\theta}_5$ con $\gamma=7$ y $T=0.0064$ seg. (Señal senoidal de referencia). Se graficó con el método de memoria de expansión.

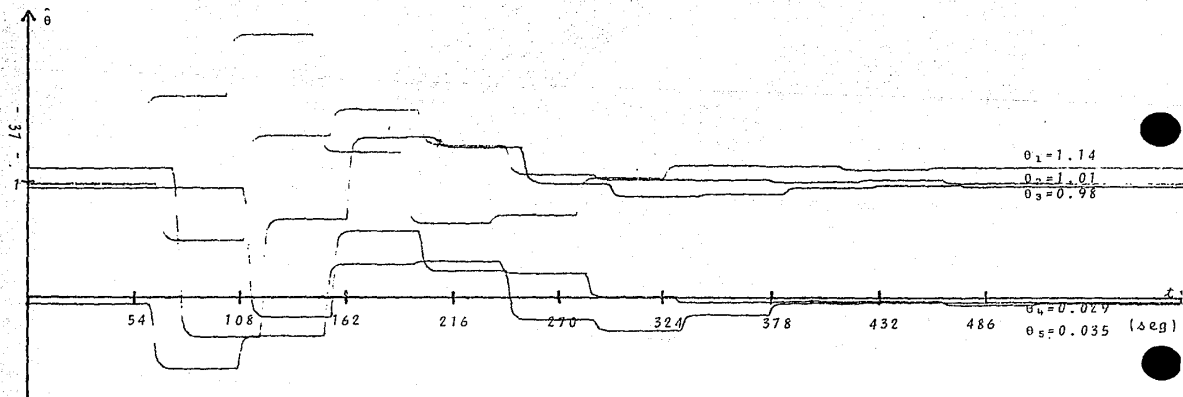
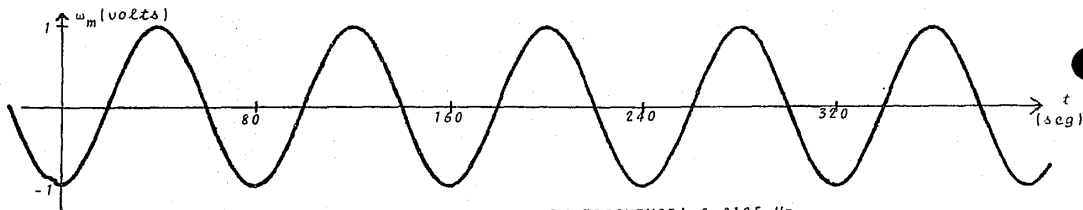
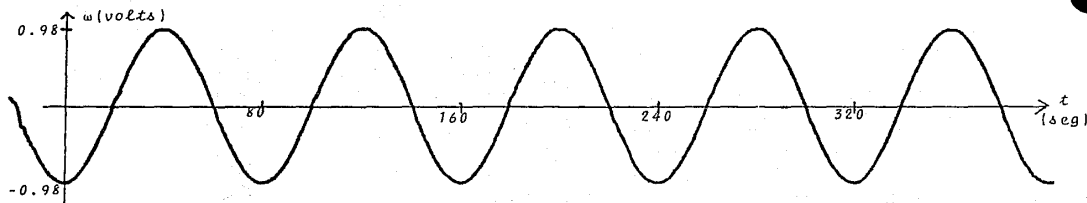


Fig. 2.12. Variación de los parámetros θ_1 , θ_2 , θ_3 , θ_4 y θ_5 con $\gamma=7$ y $T=0.0064$ seg. (Señal senoidal de referencia). Se graficaron los valores tomados después de cada ciclo de la señal ω durante 10 ciclos.



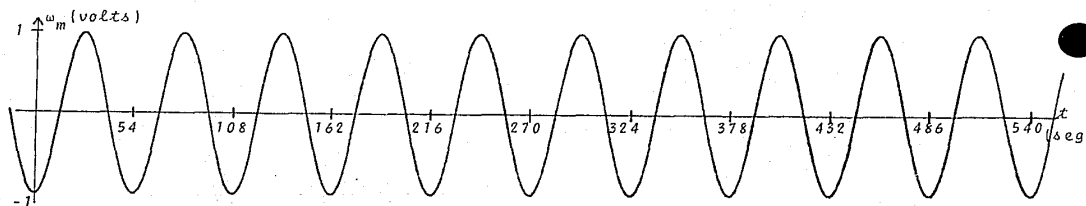
SEÑAL DE REFERENCIA ω_m DE FRECUENCIA 0.0125 Hz.

$T=0.0064$ seg.



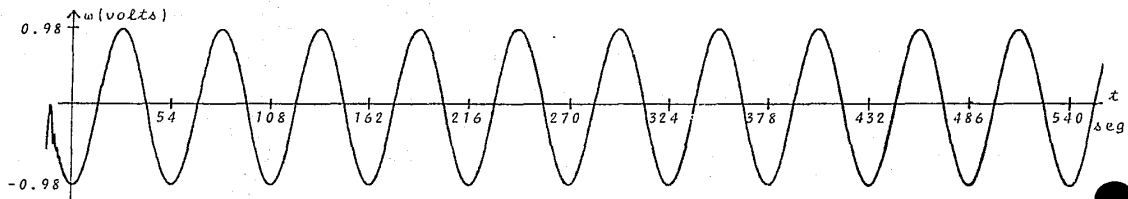
SALIDA DEL SIMULADOR ω CON $\gamma=7$

Fig. 2.13.



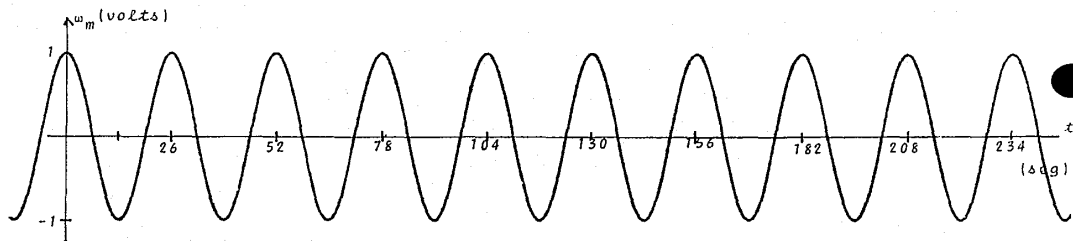
SEÑAL DE REFERENCIA ω_m DE FRECUENCIA 0.0185 Hz.

$T = 0.0064$ seg.



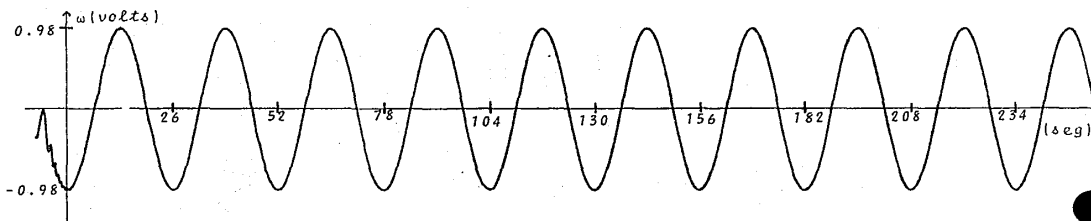
SALIDA DEL SIMULADOR ω CON $\gamma = 7$

Fig. 2.14.



SEÑAL DE REFERENCIA ω_m DE FRECUENCIA 0.0384 Hz.

$T=0.0064$ seg.



SALIDA DEL SIMULADOR ω CON $\gamma=7$

Fig. 2.15

CAPITULO 3.- CONTROL DEL MOTOR

3.1.- RESULTADOS

Como se ha mencionado, el modelo dinámico del motor de C.D. presenta características no lineales y sus parámetros son funciones de variables externas, como la inercia.

El efecto de estos factores se puede observar en la respuesta en malla abierta del motor a una entrada senoidal. Por ejemplo, graficando esta respuesta para señales senoidales de frecuencias 0.0185 Hz., 0.04 Hz., 0.076 Hz. y 0.5 Hz. (figs. 3.2, 3.3, 3.4 y 3.5, respectivamente), vemos que se presenta una zona muerta. Esta zona disminuye a medida que la frecuencia aumenta, sin embargo, la señal se distorsiona. El diagrama de conexiones para este experimento se muestra en la fig. 3.6.

Utilizando el control adaptable se espera que la no linealidad del par de fricción así como la variación del momento de inercia sean compensadas.

En este capítulo se presentan los resultados experimentales obtenidos al controlar adaptablemente el motor de C.D. Dichos resultados no fueron, sin embargo, los esperados. Una de las explicaciones probables es el hecho de que el modelo dinámico del motor fue simplificado para realizar el diseño del algoritmo de control. Algunas otras posibles causas serán comentadas posteriormente.

Al controlar el modelo lineal simulado del motor se validó el algoritmo de control provisto de la estimación de cinco parámetros considerando una señal senoidal como señal de referencia. (Programa 3 apéndice B). Por tal motivo el control del motor es llevado a cabo mediante el algoritmo del programa 3 y solo se requiere conocer la ganancia K del sistema. El diagrama de conexiones entre la interfaz analógica y el sistema servomodular (descrito en el apéndice A) es el siguiente. Fig. 3.1.

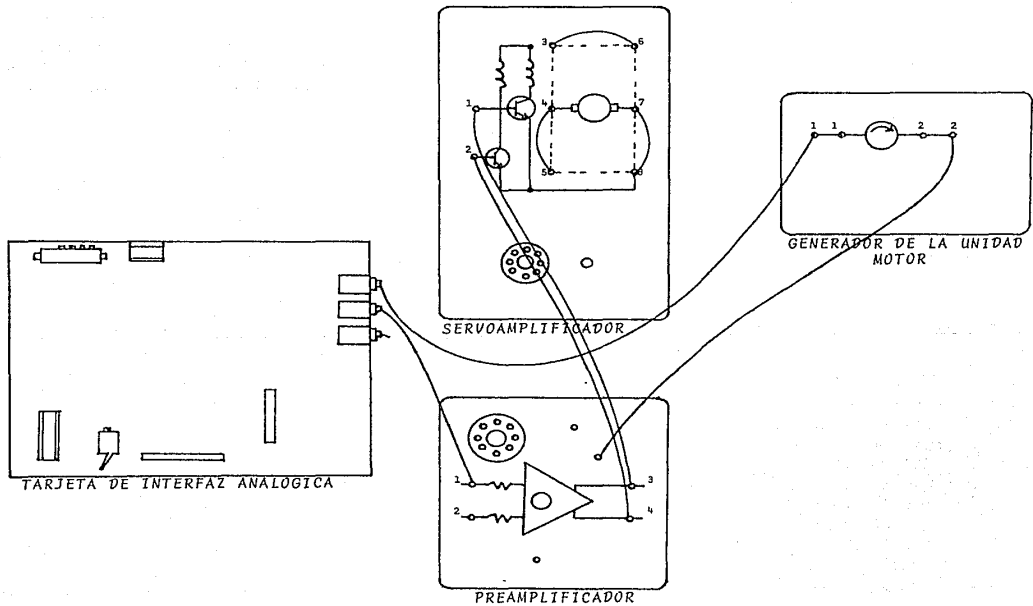


Fig. 3.1. Diagrama de conexiones entre la interfaz analógica y el sistema servomodular MS150

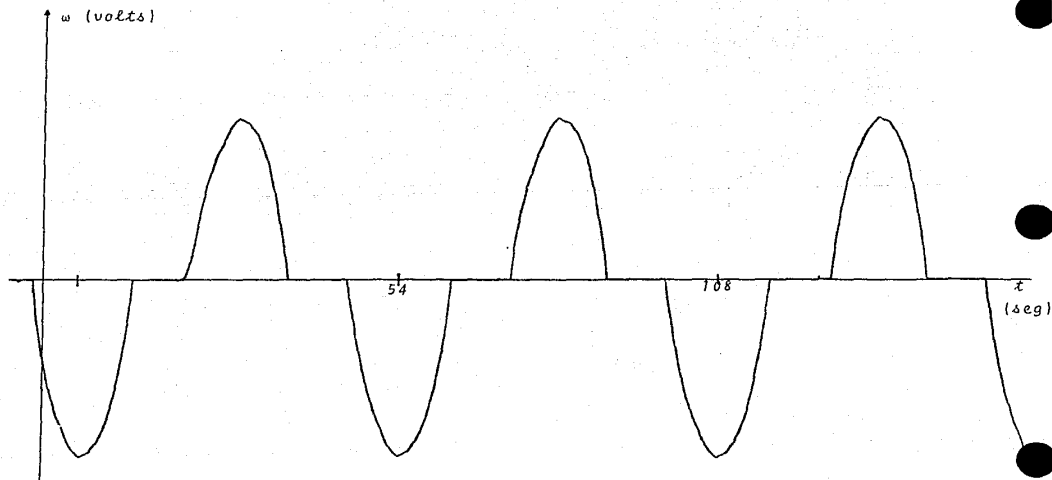


Fig. 3.2. Salida del motor, en malla abierta, ω
 $f_0 = 0.0185$ Hz.

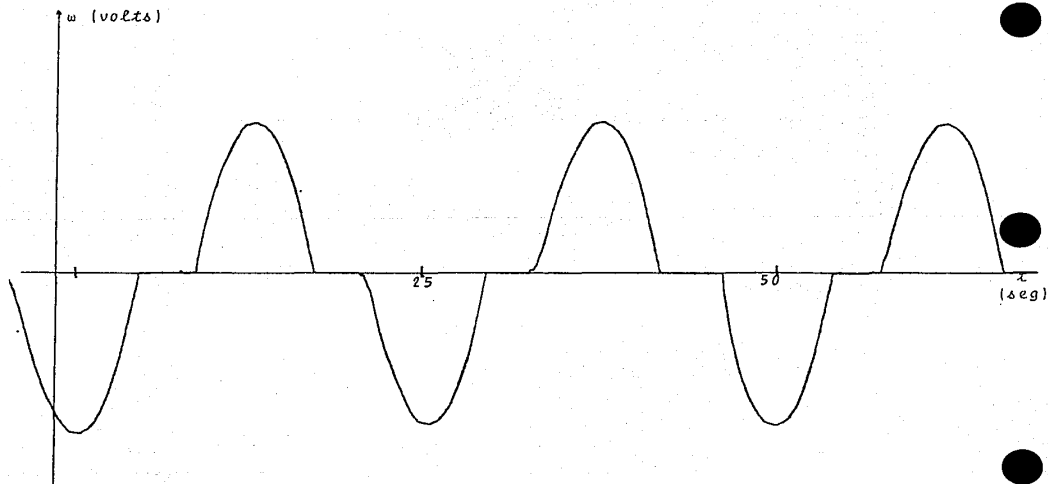


Fig. 3.3. Salida del motor, en malla abierta, ω
 $\delta = 0.04$ Hz.

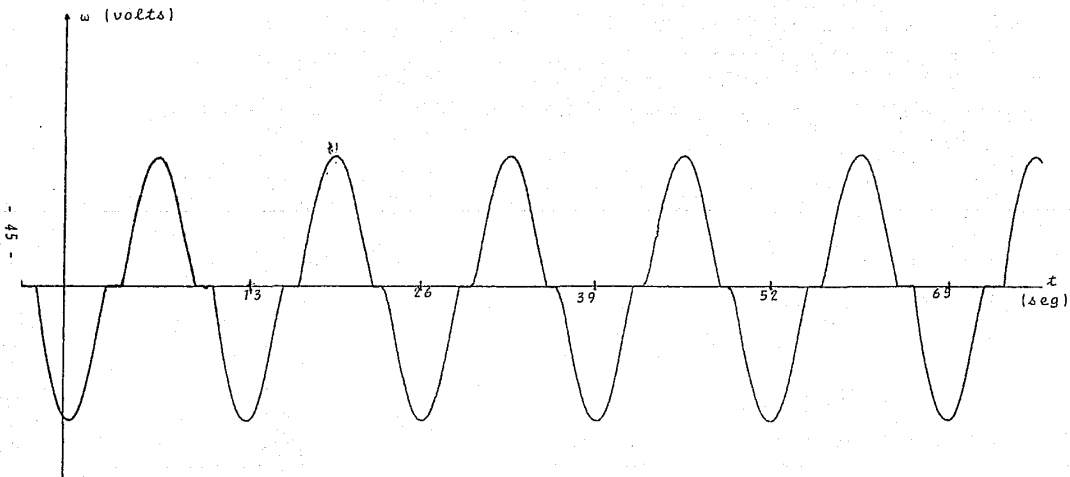


Fig. 3.4. Salida del motor, en malla abierta, ω
 $f = 0.076$ Hz.

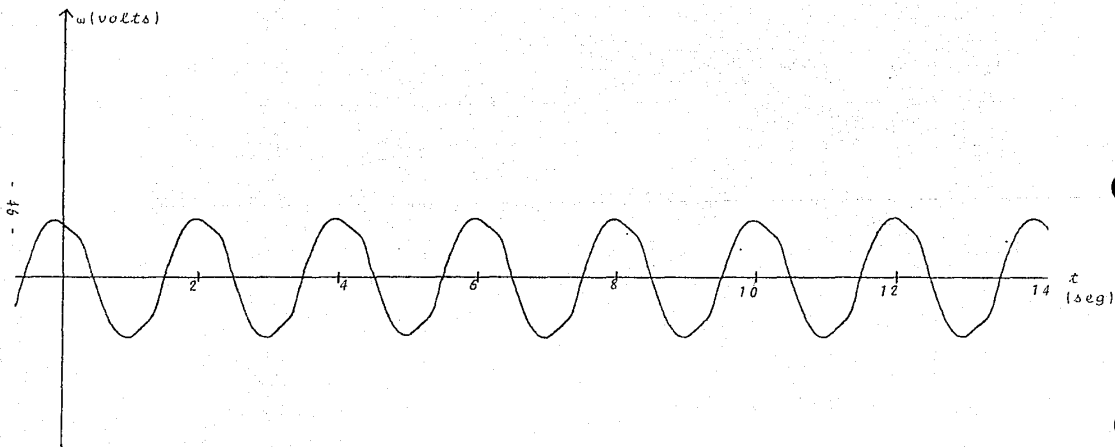
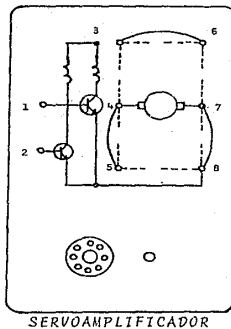


Fig. 3.5. Salida del motor, en malla abierta, w
 $f_0 = 0.5$ Hz.

GENERADOR DE
FUNCIONES

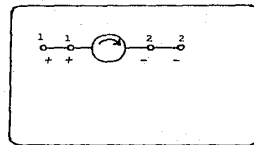
Se conecta a la en-
trada 1 del servo-
amplificador



SERVOAMPLIFICADOR

OSCILOSCOPIO

Se conecta a
1 y 2 del
generador



GENERADOR DE LA UNIDAD
MOTOR

Fig. 3.6. Diagrama de conexiones para obtener la respuesta en malla abierta del motor a una entrada senoidal

Debido a que se trabaja con señales alternas se incluye el módulo preamplificador. Por tanto, la ganancia K del sistema es el producto de la ganancia del preamplificador y la ganancia estática del motor.

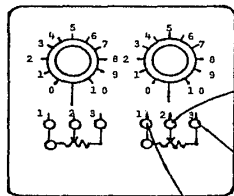
Para calcular la ganancia del motor se obtiene la característica de ve locidad, en volts, contra voltaje de alimentación, del motor controlado por armadura. Para este experimento se utilizaron 4 elementos del sistema MS150: el servocamplificador, la unidad atenuadora, la unidad motor y la fuente de poder. En la fig. 3.7 se muestra su diagrama de conexiones.

Para realizar el control por armadura, se conectan las terminales - - (3,6), (4,5) y (7,8) del servoamplificador. De la unidad atenuadora se toma el voltaje de entrada; este voltaje debe variar entre 0 y 15 volts. El voltaje de salida, que es proporcional a la velocidad del motor, se toma del - generador de la unidad motor. El voltaje de entrada se mide del cursor del potenciómetro a tierra, de la unidad atenuadora y el voltaje de salida, de las terminales 1 y 2 del generador.

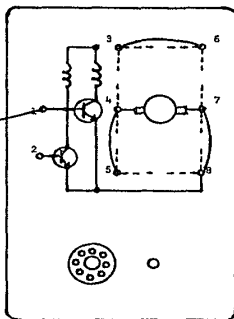
Tenemos entonces un sistema de lazo abierto en el cual V_e es el voltaje de entrada, K , la ganancia estática del motor y V_s , la velocidad de salida, en volts.

Variando V_e se obtienen los siguientes resultados:

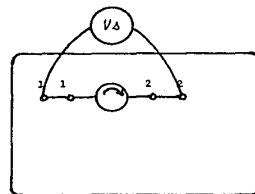
V_e (volts)	V_s (volts)
1	0
2	0
3	0
3.8	0.45
4	1.25
5	3.4
6	5.4
7	7.2
8	8.8
9	10.4



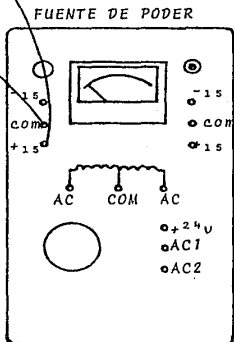
UNIDAD ATENUADORA



SERVOAMPLIFICADOR



GENERADOR DE LA UNIDAD MOTOR



FUENTE DE PODER

Fig. 3.7. Diagrama de conexiones para obtener la característica de velocidad vs. voltaje de alimentación, del motor controlado por armadura.

La gráfica de V_s vs. V_e del motor se observa en la fig. 3.8.

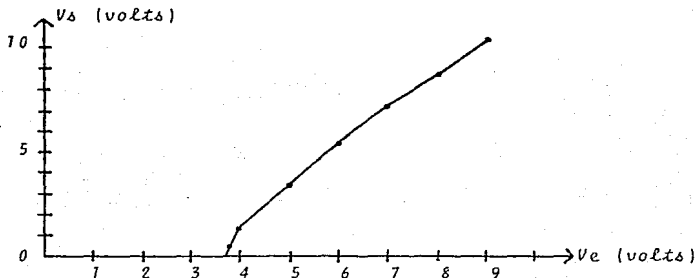


Fig. 3.8. Característica de V_s vs. V_e , del motor controlado por armadura.

De la gráfica anterior concluimos que la ganancia estática V_s/V_e no es constante y que se requiere un voltaje de 3.8 V. para que el motor gire.

Para obtener la ganancia del preamplificador se graficaron para una serie de voltajes positivos y negativos de entrada, los voltajes en las salidas 3 y 4 y se calculó la pendiente de ambas rectas. El valor obtenido es 55. Véase fig. 3.9.

3.1. Resultados

Considerando que la ganancia estática del motor no es constante, se tomó el valor de 1. Obteniéndose así una ganancia total del sistema de 55.

La variable KIN del programa deberá cargarse con el inverso de 55 en formato Q[12].

Se aplicó el algoritmo de control con una señal de referencia senoidal, ω_n de frecuencia 0.0185 Hz. y un valor de gama igual a 7, graficando la respuesta del motor, ω , fig. 3.10. Si se compara dicha respuesta con la respuesta en malla abierta para la misma frecuencia, fig. 3.2, se observa que la zona muerta disminuye considerablemente, sin embargo, aparece un ruido de alta frecuencia sobrepuesto a ω .

Analizando las posibles causas del problema, se toman en cuenta prime-

ramente las que se relacionan directamente con el control. Una de ellas es el periodo de muestreo, pues de experimentos con la planta simulada, se concluye que debe tomarse pequeño.

Se disminuyó T a 0.0032 seg. y nuevamente gama igual a 7. En este caso la zona muerta también es pequeña. El ruido se sigue presentando aunque no uniformemente pues en algunos ciclos disminuye, y después de cierto tiempo, nuevamente se presenta. Fig. 3.11.

Se graficó ω con gama igual a 1 y 3 para $T=0.0032$ seg., figs. 3.12 y 3.13, respectivamente. Así como para $T=0.0064$ seg. con los mismos valores de gama, figs. 3.14 y 3.15. Comparando estas respuestas con las figuras 3.10 y 3.11, se observa que la zona muerta aumenta y es mayor para gama igual a 1, para ambos periodos de muestreo. El ruido disminuye un poco.

Otra de las posibles causas es la no linealidad de la constante K pues no se considera en el diseño del algoritmo de control. Experimentamos variando la amplitud de la señal ω_n , así como dando otros valores a K . El ruido se sigue presentando.

Considerando que el ruido era externo, se filtró primeramente la señal de salida, ω , y posteriormente las señales de entrada y salida, i_a y ω respectivamente. Sin embargo, el ruido no desaparece.

Debido a que la señal de control debe ser muy pequeña por la alta ganancia del preamplificador se pensó que podría haber problemas con la cuantización de los convertidores por lo que se conectó a la salida de la tarjeta de interfaz un amplificador cuya ganancia es el inverso de la ganancia del preamplificador. La constante K del controlador, de esta manera, es el inverso de la ganancia estática del motor solamente. Así el valor que entra al convertidor D/A es lo suficientemente grande para que no haya ningún problema. Con esto tampoco el ruido desaparece.

También se considera como posible causa el ruido en el sistema de medición (tacómetro).

Verificando cómo se comportan los parámetros ajustables, leyéndolos directamente de las localidades de memoria de datos del TMS32010 se concluye que no tienden a un valor determinado, lo que puede ser ocasionado por el ruido que presenta la señal.

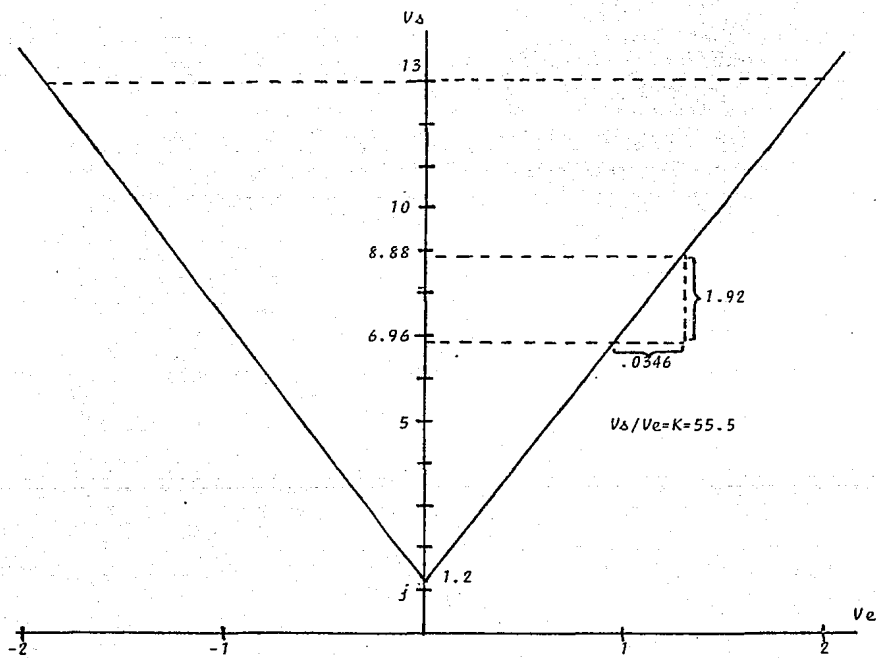


Fig. 3.9. Gráfica de V_s Vs. V_e del preamplificador.

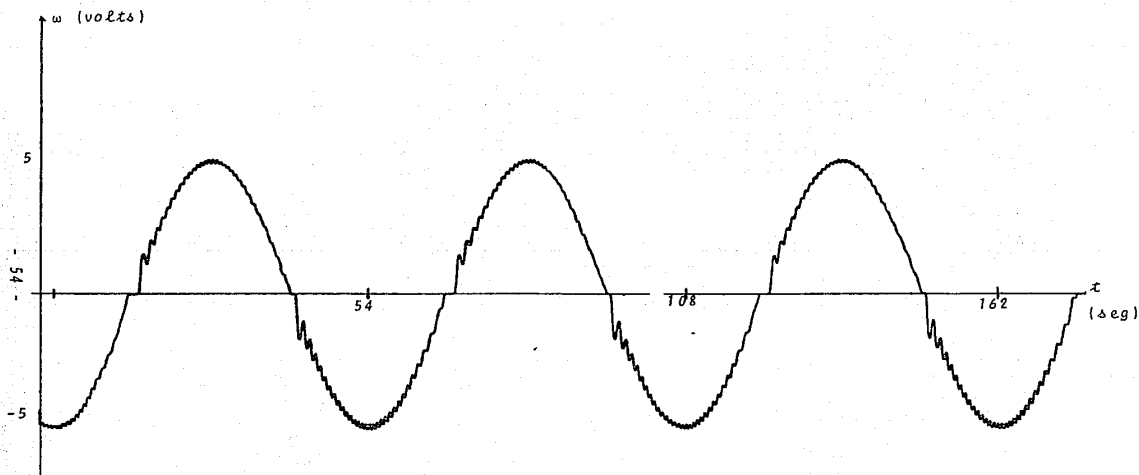


Fig. 3.10. Salida del motor, ω , con $\gamma=7$
 $T=0.0064$ seg.

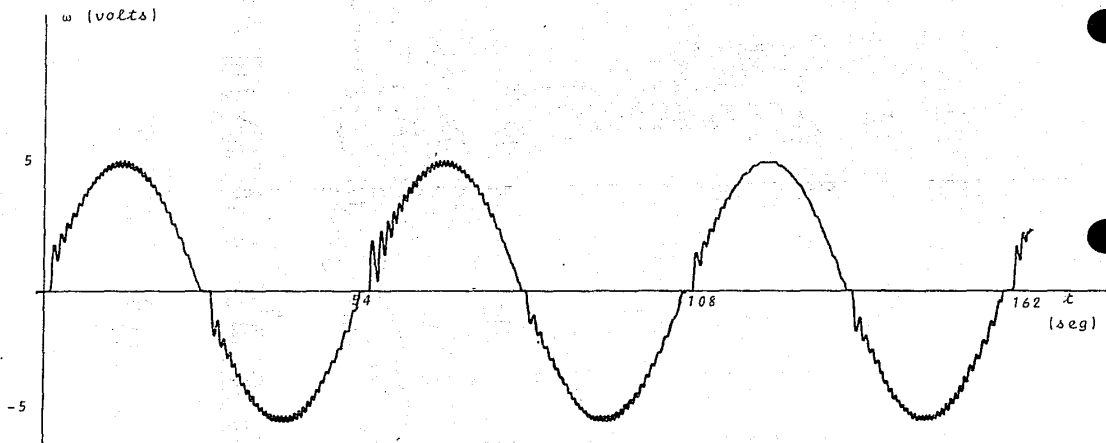


Fig. 3.11. Salida del motor, ω , con $\gamma=7$
 $T=0.0032$ seg.

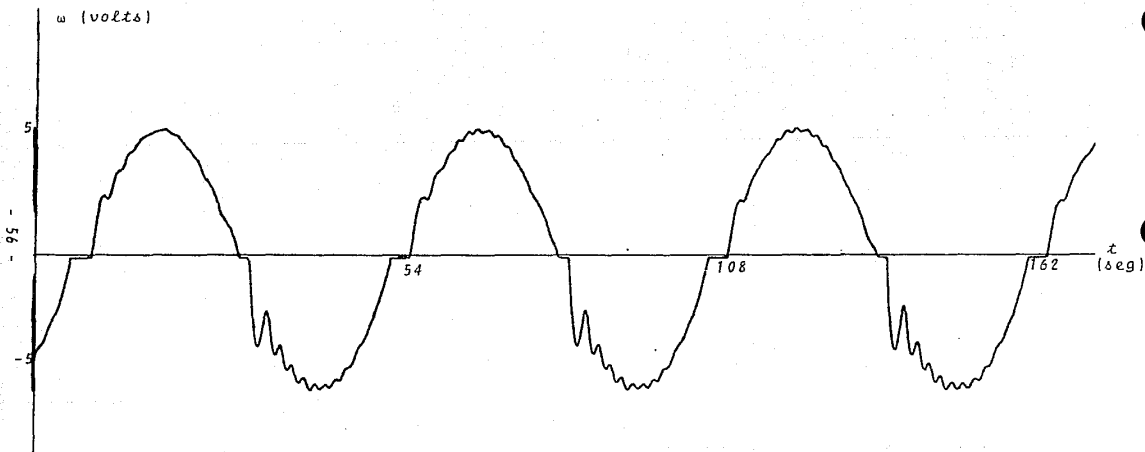


Fig. 3.12. Salida del motor, ω , con $\gamma=1$
 $T=0.0032$ seg.

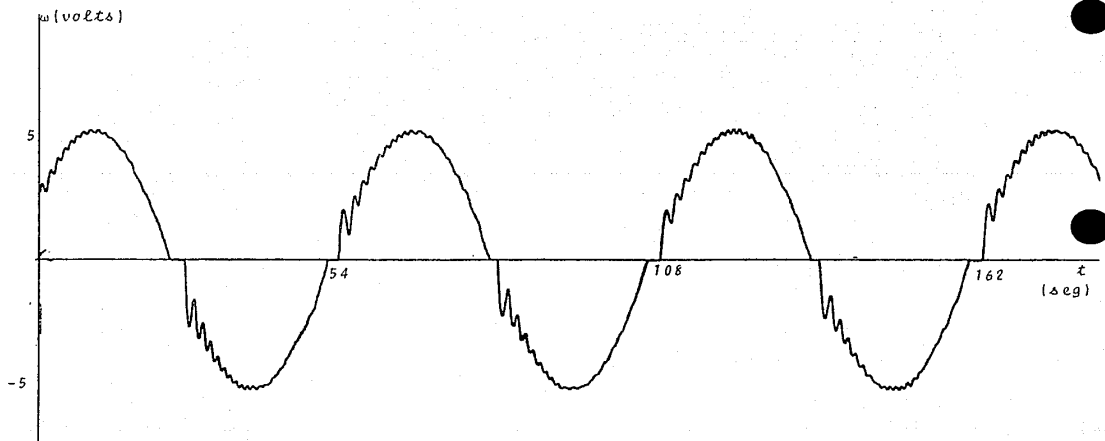


Fig. 3.13. Salida del motor, ω , con $\gamma=3$
 $T=0.0032$ seg.

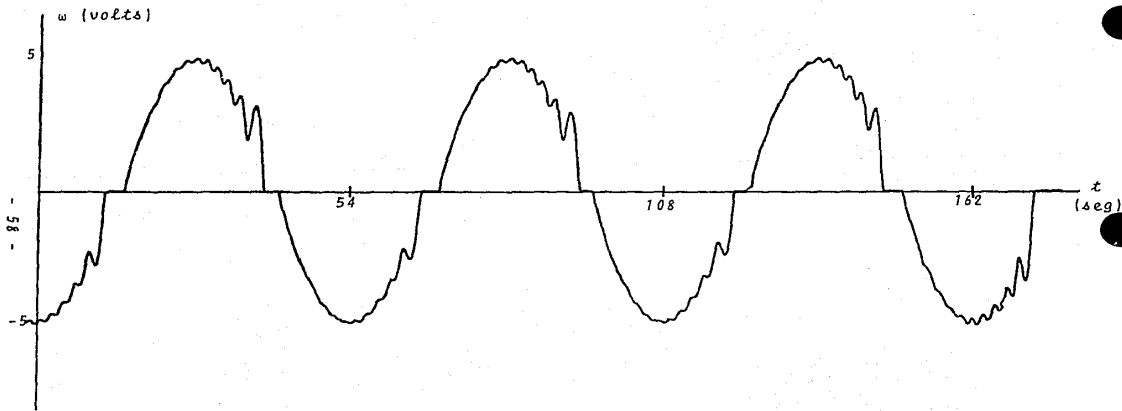


Fig. 3.14. Salida del motor, ω , con $\gamma=1$
 $T=0.0064$ seg.

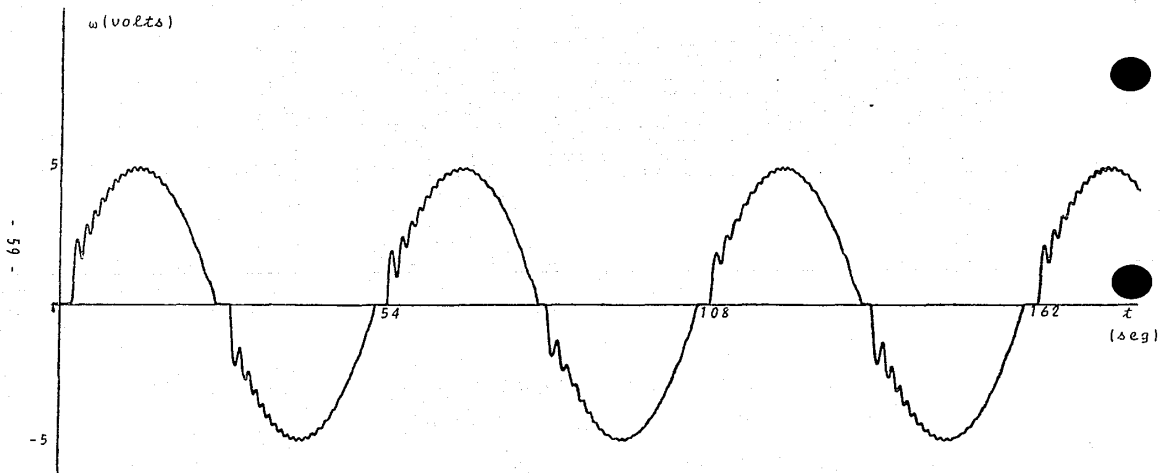


Fig. 3.15. Salida del motor, ω , con $\gamma=3$
 $T=0.0064 \Delta$ seg.

C O N C L U S I O N E S

El control de velocidad de motores de corriente directa presenta problemas debido a los efectos de la no linealidad del par de fricción así como a las variaciones del momento de inercia.

Podríamos realizar el control mediante un sistema de retroalimentación lineal, sin embargo, se requiere para su buen funcionamiento, un modelo exacto del motor, es decir, que los efectos de la fricción y cambios de inercia sean conocidos. Esto no sucede siempre en la realidad por lo que un sistema de control adaptable es planteado como posible solución ya que estos sistemas tienen la finalidad de controlar procesos físicos cuyos modelos matemáticos contienen parámetros desconocidos y variantes en el tiempo.

El objetivo de la presente tesis fue el de implementar un sistema de control adaptable, de velocidad, para motores de corriente directa empleando un microprocesador rápido. Su desarrollo consistió en las siguientes etapas:

1. Diseño y construcción de un simulador analógico para simular un modelo lineal del motor con el fin de realizar experimentos para verificar el funcionamiento del algoritmo de control.
2. Implementación del algoritmo de control en el microprocesador TMS32010 así como la elaboración de programas auxiliares para la graficación de los parámetros ajustables.
3. Aplicación del algoritmo de control a la planta simulada considerando tres casos.
 - a). Implementación con 3 parámetros y ω_n cuadrada.
 - b). Implementación con 5 parámetros y ω_n cuadrada.
 - c). Implementación con 5 parámetros y ω_n senoidal.
4. Aplicación de la implementación del controlador con 5 parámetros y -

una señal senoidal de referencia al motor de corriente directa del sistema servomodular MS150.

5. Obtención e interpretación de resultados.

De los resultados obtenidos en experimentos realizados con el modelo lineal simulado del motor se concluye que el controlador cumple con su objetivo que es el de generar una señal de control, i_a , a fin de que la respuesta de la planta, ω , tienda al modelo de referencia ω_m en un tiempo determinado.

Después de haber comprobado que el algoritmo de control funciona como estaba previsto se pasó a la implementación en el motor de C.D. del sistema servomodular MS150.

De los experimentos realizados con el motor, los resultados no son los esperados ya que la señal, ω , presenta un ruido de alta frecuencia lo que implica que el controlador no funcione correctamente.

Las posibles causas son:

- a). El modelo dinámico del motor fue simplificado para realizar el algoritmo de control.
- b). Discretización del algoritmo de control ya que la teoría fue desarrollada para un algoritmo continuo.
- c). El modelo dinámico del motor contiene otras características no lineales no consideradas en el diseño del algoritmo de control, como la ganancia estática, la cual depende de su punto de operación.
- d). Ruido en el sistema de medición (tacómetro).
- e). Ruido externo.

R E F E R E N C I A S

1. OGATA KATSUHIKO, "Ingeniería de control moderna".
Prentice-Hall Hispanoamericana S.A.
1986.
2. CANALES RUIZ ROBERTO, "Análisis de sistemas dinámicos y control automático".
Limusa, 1980.
3. BUITRON SANCHEZ HORACIO, "Operación, control y protección de motores eléctricos".
H.P. Editor de libros técnicos.
1984.
4. MANUAL DEL SISTEMA SERVOMODULAR MS150, Reporte CINVESTAV.
1986.
5. ARAGON POMPILIO, KELLY RAFAEL, ORTEGA ROMEO, "A Robustly stable adaptive compensator of friction and load in D.C. motors".
Reporte interno DEEFI, 1987.
6. ASTROM KARL JOHAN, "Adaptive Feedback Control", Proceedings of the IEEE, vol 75, No.2, 1987.
7. ASTROM KARL JOHAN, "Theory and applications of adaptive control", Automática, September 1983.
8. KELLY RAFAEL, "A linear state feedback plus adaptive feed - forward control for D.C. servomotors", IEEE Trans. on Industrial Electronics, vol IE-34, No. 2, may 1987.

APENDICE A

1. - DESCRIPCION DEL SIMULADOR DE F.T.
2. - DESCRIPCION DEL SISTEMA SERVOMODULAR MS150
3. - DESCRIPCION GENERAL DEL MODULO DE EVALUACION DEL TMS32010 (EVM)
4. - DESCRIPCION GENERAL DEL MICROPROCESADOR TMS32010
5. - DESCRIPCION DE LA TARJETA DE INTERFAZ ANALOGICA

A.1.- SIMULADOR DE FUNCIONES DE TRANSFERENCIA

A.1.1.- Función de transferencia

Los sistemas dinámicos (mecánicos, eléctricos, térmicos, etc.) son -- descritos mediante modelos matemáticos, que de acuerdo a las características que se quieren estudiar de ellos, se utilizará una representación de -- terminada, por ejemplo, para el análisis de respuesta transitoria de sistemas de una sola entrada y una sola salida, la función de transferencia es la representación adecuada.

El concepto de función de transferencia se aplica a sistemas lineales e invariantes en el tiempo; es una propiedad de los sistemas ya que es independiente de la entrada y se expresa como la razón de dos polinomios de variable compleja en S con coeficientes reales. [1], [2]. En general está dada por:

$$G(S) = \frac{Y(S)}{X(S)} = \frac{y_m S^m + y_{m-1} S^{m-1} + \dots + y_1 S + y_0}{x_n S^n + x_{n-1} S^{n-1} + \dots + x_1 S + x_0}$$

Cada uno de los polinomios $Y(S)$ y $X(S)$ pueden ser expresados en forma factorizada,

$$Y(S) = y_m (S - Z_1) (S - Z_2) (S - Z_3) \dots (S - Z_m)$$

$$X(S) = (S - P_1) (S - P_2) (S - P_3) \dots (S - P_n)$$

donde P_i son los polos y Z_i los ceros del sistema. Tanto los polos como los ceros pueden ser representados gráficamente en el plano complejo. Ejemplo.

$$G(S) = \frac{(S+1)(S-3)}{(S+1-j)(S+1+j)(S+2)}$$

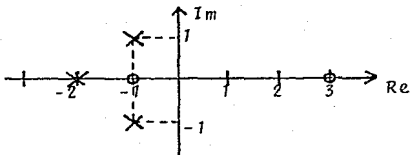


Fig. A.1.1. Patrón de polos y ceros

De acuerdo al orden del polinomio del denominador se establece el orden del sistema.

A.1.2. Sistema de primer orden

La relación entrada/salida (función de transferencia) de un sistema de primer orden está dada por,

$$H(S) = \frac{Y(S)}{X(S)} = \frac{1}{\tau S + 1}$$

Analizando la respuesta escalón de dicho sistema se tiene,

$$Y(S) = H(S)X(S)$$

donde $X(S) = 1/S$ que es la transformada de Laplace de la función escalón unitario.

Sustituyendo,

$$Y(S) = \frac{1}{\tau S + 1} \frac{1}{S}$$

Dividiendo y desarrollando en fracciones parciales,

$$Y(S) = \frac{1/\tau}{S(S + 1/\tau)} = \frac{A}{S} + \frac{B}{S + 1/\tau}$$

$$A = 1 \quad B = -1$$

$$Y(S) = \frac{1}{S} - \frac{1}{S + 1/\tau}$$

Por transformada inversa se obtiene que,

$$y(t) = u_{-1}(t) - e^{-(1/\tau)t} \quad \text{para } t \geq 0$$

donde

u_{-1} es un escalón unitario

A continuación se muestra la curva de la respuesta escalón de un sistema de primer orden, descrita para la ecuación anterior.

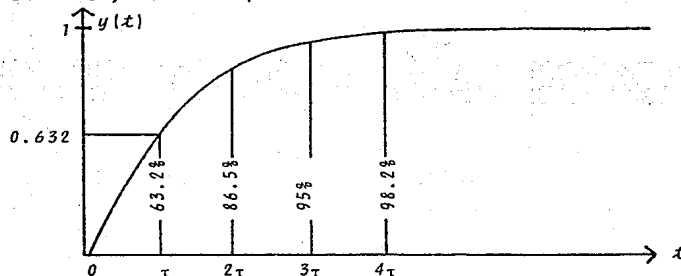


Fig. A.1.2. Respuesta escalón de un sistema de primer orden

La característica más importante que se estudia de esta curva es el valor de τ , la constante de tiempo, que es el tiempo para el cual $y(t)$ ha alcanzado el 63.2% de su variación total. Cuanto más pequeña sea la constante de tiempo, más rápida es la respuesta del sistema. [1].

A.1.3. Sistema de segundo orden

Estos sistemas tienen una función de transferencia de la forma,

$$G(S) = \frac{1}{(S+\alpha)^2 + \omega^2}$$

en la que $G(S)$ puede escribirse como,

$$G(S) = \frac{1}{S + 2\xi\omega_n S + \omega_n^2}$$

donde

$$\xi\omega_n = \alpha$$

$$\omega_n^2 = \alpha^2 + \omega^2$$

α , el amortiguamiento real

ω , la frecuencia real

ξ , el amortiguamiento relativo o coeficiente de amortiguamiento

ω_n , la frecuencia natural

Los polos del sistema están dados en función de ξ y ω_n , por,

$$P_1 = -\xi\omega_n + j\omega_n\sqrt{1-\xi^2}$$

$$P_2 = -\xi\omega_n - j\omega_n\sqrt{1-\xi^2}$$

Gráficamente,

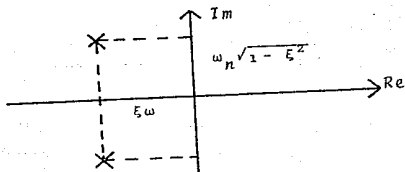


Fig. A.1.3. Patrón de polos y ceros de un sistema de segundo orden. De acuerdo a los valores de ξ , se tienen cuatro tipos de respuesta. -

Respuesta:

1. Subamortiguada. $0 < \xi < 1$. En este caso los dos polos son complejos.
2. Críticamente amortiguada. $\xi = 1$. Los dos polos coinciden y son reales.
3. Sobreamortiguada. $\xi > 1$. Los polos son reales pero no coinciden.
4. No amortiguada. $\xi = 0$. Los polos están sobre el eje imaginario.

Respuesta escalón de un sistema de segundo orden para diversos valores de ξ .

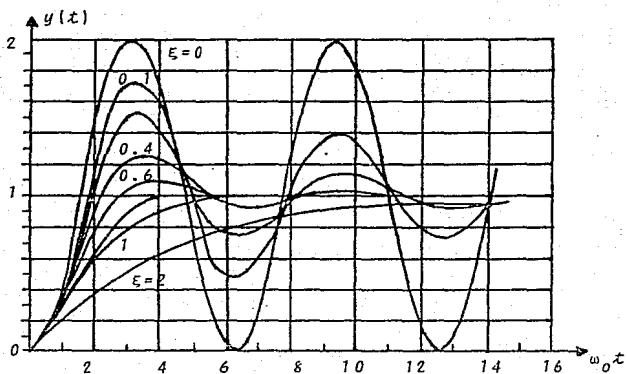


Fig. A.1.4. Respuesta escalón de un sistema de segundo orden

Se observa que las características del sistema dependen del valor de ξ .

ξ .

Las especificaciones más importantes de la respuesta transitoria son:

1. Sobrepasso (M_p), que es la máxima sobredesviación con respecto al valor final de la respuesta del sistema; se expresa en porcentaje.

A pequeños valores de ξ el sobrepasso es mayor.

$$M_p = e^{-\frac{\xi \pi}{\sqrt{1-\xi^2}}}$$

- 2.- Tiempo de retardo (t_r), es el tiempo en el cual la respuesta alcanza por primera vez el 50% de su valor final.
- 3.- Tiempo de levantamiento (t_l), es el tiempo que le toma al sistema para que la respuesta pase del 10% al 90% del valor final.

$$t_l = \frac{\pi - \phi}{\omega_n \sqrt{1-\xi^2}} \quad \phi = \cos^{-1} \xi$$

- 4.- Tiempo de asentamiento (t_a), es el tiempo al cual la respuesta está dentro del 5% de su valor final.

$$t_a = \frac{5}{\xi \omega_n}$$

Lo anterior se muestra en la figura.

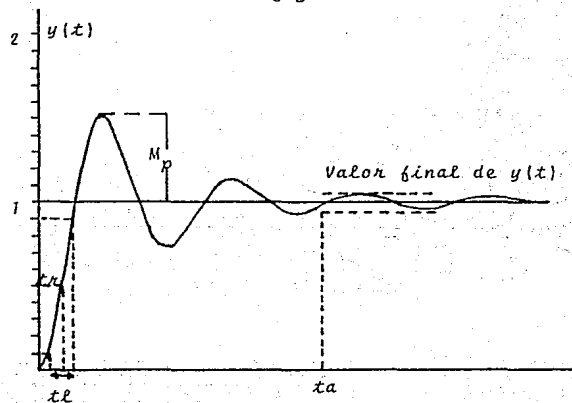
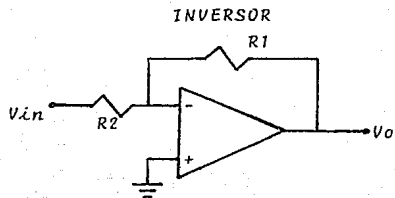


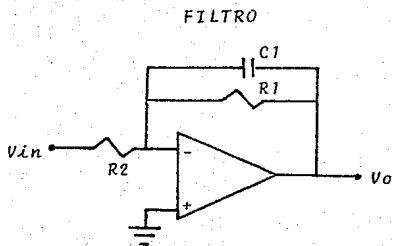
Fig. A.1.5. Especificaciones de la respuesta transitoria

A.1.4. Descripción del simulador de F.T.

El simulador de funciones de transferencia nos da una variedad de funciones de primero y segundo orden. Los circuitos típicos que lo constituyen son: el inversor, el integrador y el filtro.

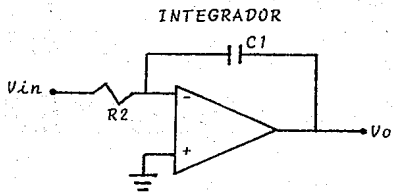


$$G(S) = \frac{V_o}{V_{i.n}} = - \frac{R1}{R2}$$



$$G(S) = \frac{V_o}{V_{i.n}} = - \frac{R1/R2}{R1C1S + 1}$$

donde $R1C1 = \tau$ y $R1/R2 = K$



$$G(S) = - \frac{1}{R2C1S}$$

Fig. A.1.6. Circuitos típicos que constituyen el simulador de F.T.

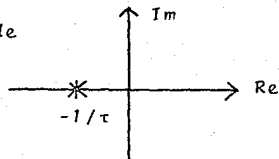
A.1.5. Función de transferencia de primer orden

Los amplificadores A1 y A2 simulan una función de transferencia de primer orden.

$$G(S) = - \frac{K}{1 + \tau S}$$

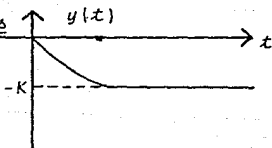
En este caso, el patrón de polos y ceros es,

Fig. A.1.7. Patrón de polos y ceros



y, la curva de la respuesta $y(t)$, a una entrada escalón unitario,

Fig. A.1.8. Respuesta escalón

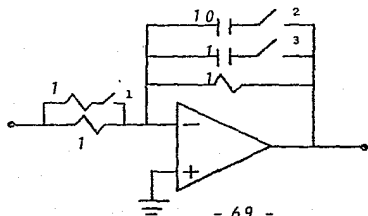


La ganancia K de A1 toma los valores de 1 y 2 y la constante de tiempo, 0, 1 y 10, es decir, para $\tau=0$, A1 funciona como inversor.

La ganancia de A2 toma valores de 0.5 y 1 y al igual que el anterior la constante de tiempo, 0, 1 y 10.

Las funciones que se obtienen de estos dos amplificadores se indican en las tablas siguientes, especificando la posición que debe tener cada interruptor en cada caso.

AMPLIFICADOR 1

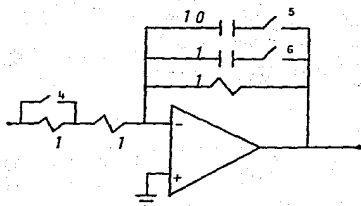


SW1	SW2	SW3	FUNC. TRANS.
A	A	A	- 1
C	A	A	- 2
A	C	A	$-\frac{1}{10S + 1}$
C	C	A	$-\frac{2}{10S + 1}$
A	A	C	$-\frac{1}{S + 1}$
C	A	C	$-\frac{2}{S + 1}$

A (abierto).

C (cerrado)

AMPLIFICADOR 2



SW4 SW5 SW6 FUNC. TRANS.

A	A	A	- 0.5
C	A	A	- 1
A	C	A	- $\frac{0.5}{TOS + 1}$
C	C	A	- $\frac{1}{TOS + 1}$
A	A	C	- $\frac{0.5}{S + 1}$
C	A	C	- $\frac{1}{S + 1}$

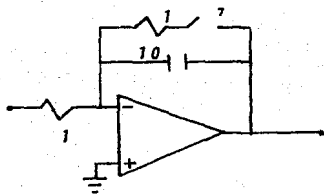
A (abierto)

C (cerrado)

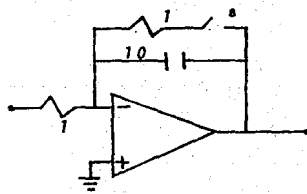
ESTA TESIS NO PUEDE
SALIR DE LA BIBLIOTECA

En lo que respecta a los amplificadores A3 y A4, Estos funcionan como integradores y también simulan una función de transferencia de primer orden.

AMPLIFICADOR 3



AMPLIFICADOR 4



SW7	FUNC. TRANS.	SW8	FUNC. TRANS.
A	$-\frac{1}{TOS}$	A	$-\frac{1}{TOS}$
C	$-\frac{1}{TOS + T}$	C	$-\frac{1}{TOS + T}$

A.1.6. Función de transferencia de segundo orden

Conectando los amplificadores A1 y A2 en cascada, se tienen las funciones de transferencia de segundo orden.

De todas las combinaciones posibles, que se muestran a continuación se encuentran dos tipos de respuesta; la respuesta sobreamortiguada y la críticamente amortiguada.

FUNCIONES DE TRANSFERENCIA DE SEGUNDO ORDEN CONECTANDO A1 Y A2 EN CASCADEA

0.5	1	$\frac{0.5}{s+1}$	$\frac{1}{s+1}$	$\frac{0.5}{10s+1}$	$\frac{1}{10s+1}$
1	2	$\frac{1}{s+1}$	$\frac{2}{s+1}$	$\frac{1}{10s+1}$	$\frac{2}{10s+1}$
$\frac{0.5}{s+1}$	$\frac{1}{s+1}$	$\frac{0.5}{(s+1)(s+1)}$	$\frac{1}{(s+1)(s+1)}$	$\frac{0.5}{(s+1)(10s+1)}$	$\frac{1}{(s+1)(10s+1)}$
$\frac{1}{s+1}$	$\frac{2}{s+1}$	$\frac{1}{(s+1)(s+1)}$	$\frac{2}{(s+1)(s+1)}$	$\frac{1}{(s+1)(10s+1)}$	$\frac{2}{(s+1)(10s+1)}$
$\frac{0.5}{10s+1}$	$\frac{1}{10s+1}$	$\frac{0.5}{(10s+1)(s+1)}$	$\frac{1}{(10s+1)(s+1)}$	$\frac{0.5}{(10s+1)(10s+1)}$	$\frac{1}{(10s+1)(10s+1)}$
$\frac{1}{10s+1}$	$\frac{2}{10s+1}$	$\frac{1}{(10s+1)(s+1)}$	$\frac{2}{(10s+1)(s+1)}$	$\frac{1}{(10s+1)(10s+1)}$	$\frac{2}{(10s+1)(10s+1)}$

Tomemos como primer ejemplo,

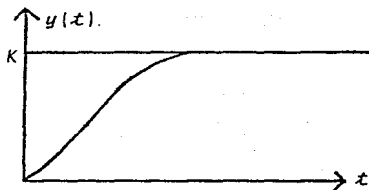
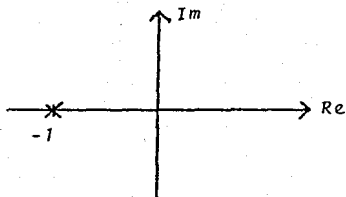
$$1. \quad G(S) = \frac{K}{(S+1)(S+1)}$$

$$G(S) = \frac{K}{S^2 + 2S + 1}$$

y comparando con la expresión general,

$$H(S) = \frac{K}{S^2 + 2S + 1} = \frac{K}{S^2 + 2\xi\omega_n S + \omega_n^2}$$

ξ y ω_n tienen el valor de 1. Por tanto, se trata de un sistema, de acuerdo al valor de ξ , críticamente amortiguado, cuyos polos son iguales y reales.



g. A.1.9 Patrón de polos y ceros

Respuesta escalón

La respuesta escalón se obtiene mediante el desarrollo en fracciones parciales,

$$Y(S) = \frac{K}{(S+1)(S+1)S}$$

$$Y(S) = \frac{A}{S} + \frac{B}{S+1} + \frac{C}{(S+1)^2}$$

$$A = K \quad B = -K \quad C = -K$$

$$Y(S) = \frac{K}{S} - \frac{K}{S+1} - \frac{K}{(S+1)^2}$$

$$y(t) = K u_{-1}(t) - K e^{-t} - K t e^{-t}$$

Otra de las funciones de transferencia es,

$$2. \quad G(S) = \frac{K}{(10S+1)(10S+1)}$$

$$G(S) = \frac{K}{100S^2 + 20S + 1}$$

También tiene una respuesta críticamente amortiguada pues $\xi=1$. Sin embargo, la frecuencia natural es de 0.1.

La respuesta escalón es,

$$y(t) = Kv_{-1}(t) - Ke^{-.1t} - 0.1Kte^{-.1t}$$

La respuesta sobreamortiguada nos la da una función de transferencia de la forma,

$$3. G(S) = \frac{K}{(10S + 1)(S + 1)}$$

$$G(S) = \frac{K}{S^2 + 1.1S + 0.1}$$

ya que $\xi=1.739$ ($\xi > 1$) y $\omega_n=0.3162$, los polos son reales pero no coinciden.

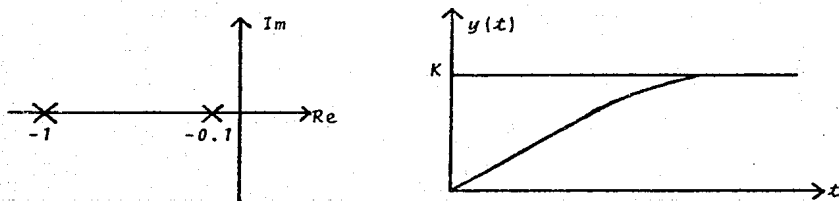


Fig. A.1.10 Patrón de polos y ceros Respuesta escalón.

La respuesta escalón está dada por,

$$y(t) = Kv_{-1}(t) - 1.1Ke^{-1t} - 0.11Ke^{-.1t}$$

En cuanto a los amplificadores A3 y A4, el sistema de segundo orden resultante puede ser realimentado o no.

El bloque K de realimentación toma los valores de 0, 1 ó 4. Haciendo notar que el valor de 0 indica que el sistema no se encuentra realimentado.

Las funciones de transferencia obtenidas cuando $K=0$ son:

A3	A4	FUNC. TRANS.
Intg.	Intg.	a. $\frac{i}{100S^2}$
Intg.	Filtro	b. $\frac{1}{10S(10S+1)}$
Filtro	Filtro	c. $\frac{1}{(10S+1)^2}$

El patrón de polos y ceros de cada una de ellas se muestra a continuación.

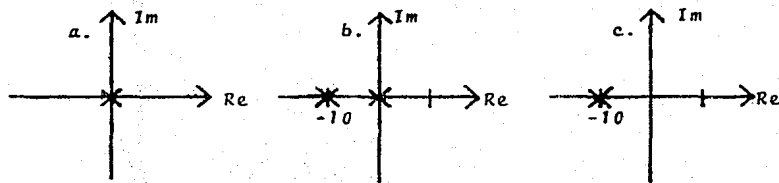


Fig. A.1.11 Patrón de polos y ceros de las F.T. de la tabla anterior

Para todo K , la función de transferencia del sistema realimentado

$$H(S) = \frac{G(S)}{1 + KG(S)}$$

se deduce a partir del siguiente diagrama.

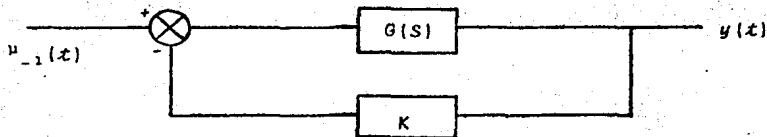


Fig. A.1.12. Diagrama de bloques de un sistema realimentado

Realimentando la función,

$$G(S) = \frac{1}{i\omega S^2}$$

Cuando $K=1$

$$H(S) = \frac{0.01}{S^2 + 0.01}$$

donde

$$\xi = 0$$

$$\omega_n = 0.1$$

por tanto, es un sistema no amortiguado

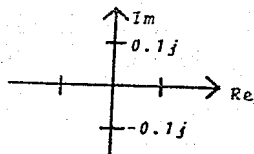


Fig. A.1.13 Patrón de polos y ceros

Desarrollando para obtener la respuesta escalón,

$$Y(S) = \frac{0.01}{S(S^2 + 0.01)}$$

en fracciones parciales,

$$Y(S) = \frac{A}{S} + \frac{B + CS}{S^2 + 0.01}$$

$$A = 1$$

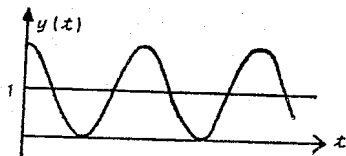
$$B = 0 \quad C = -1$$

$$Y(S) = \frac{1}{S} + \frac{S}{S^2 + 0.01}$$

resulta una respuesta de la forma,

$$y(t) = u_{-1}(t) - \cos 0.1t$$

Fig. A.1.14 Respuesta escalón



Cuando $K=4$

$$H(S) = \frac{0.01}{S^2 + 0.04}$$

$$\xi = 0$$

$$y \quad \omega_n = 0.2$$

También se trata de un sistema no amortiguado cuyo patrón de polos y ceros es:

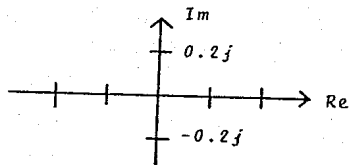


Fig. A.1.15. Patrón de polos y ceros

La respuesta escalón se muestra en la siguiente figura y se obtiene del desarrollo,

$$y(S) = \frac{0.01}{S(S^2 + 0.04)} = \frac{A}{S} + \frac{B + CS}{S^2 + 0.04}$$

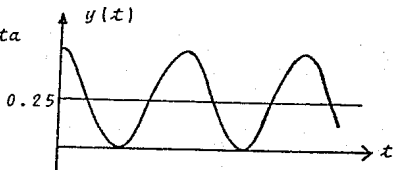
$$A = 0.25 \quad B = 0 \quad C = -0.25$$

$$y(S) = \frac{0.25}{S} - \frac{0.25S}{S^2 + 0.04}$$

finalmente,

$$y(t) = 0.25u_{-1}(t) - 0.25 \cos 0.2t$$

Fig. A.1.16. Respuesta escalón



Realimentando ahora,

$$G(S) = \frac{1}{100S^2 + 10S}$$

Cuando $K=1$

$$H(S) = \frac{.01}{S^2 + 0.1S + 0.01}$$

Los valores de ξ y ω_n son 0.5 y 0.1 respectivamente; en este caso se trata de un sistema subamortiguado.

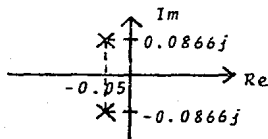


Fig. A.1.17 Patrón de polos y ceros

realizando la transformada inversa para obtener la respuesta escalón,

$$Y(S) = \frac{0.01}{S[(S+0.05)^2 + .0075]}$$

$$Y(S) = \frac{A}{S} + \frac{B + CS}{(S+0.05)^2 + 0.0075}$$

$$A = 1 \quad B = -0.1 \quad C = -1$$

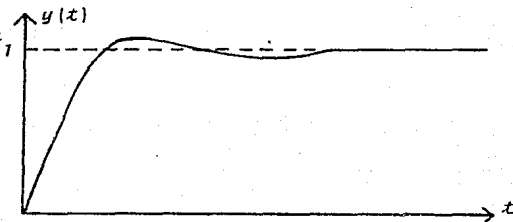
$$Y(S) = \frac{1}{S} - \frac{0.1}{(S+0.05)^2 + 0.0075} - \frac{S}{(S+0.05)^2 + 0.0075}$$

Por tanto,

$$y(t) = u_{-1} - 1.15e^{-.05t} \text{ sen } 0.0866t - e^{-.05t} \text{ cos } 0.0866t$$

La curva de la respuesta que se describe en la ecuación anterior es,

Fig. A.1.18 Respuesta escalón



Para $K=4$

Para

$$H(S) = \frac{0.01}{S^2 + 0.1S + 0.04}$$

Los valores de ξ y ω_n son 0.25 y 0.2; por tanto, es un sistema subamortiguado.

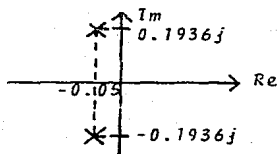


Fig. A.1.19 Patrón de polos y ceros

De transformada inversa,

$$Y(S) = \frac{0.01}{S((S+0.05)^2 + 0.375)} = \frac{A}{S} + \frac{B + CS}{(S+0.05)^2 + 0.375}$$

$$A = 0.25$$

$$B = -0.025$$

$$C = -0.25$$

$$Y(S) = \frac{0.25}{S} - \frac{0.025}{(S+0.05)^2 + 0.375} - \frac{0.25S}{(S+0.05)^2 + 0.375}$$

La respuesta escalón es,

$$y(t) = 0.25\mu_{-1}(t) - 0.667e^{-0.05t} \text{sen } 0.1936t - 0.25e^{-0.05t} \text{cos } 0.1936t$$

cuya curva es,

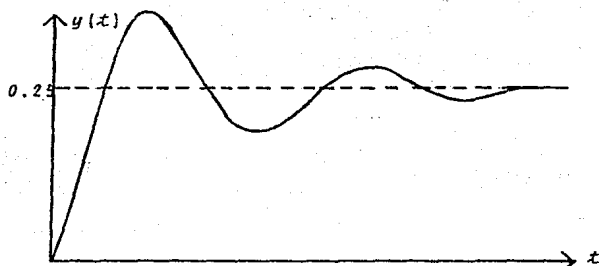


Fig. A.1.20 Respuesta escalón

Realimentando la función,

$$G(S) = \frac{1}{100S^2 + 20S + 1}$$

Para $K=1$

$$H(S) = \frac{0.01}{S^2 + 0.2S + 0.02}$$

Los valores de ξ y ω_n son 0.7071 y $\sqrt{0.02}$, lo que indica que también es un sistema subamortiguado.

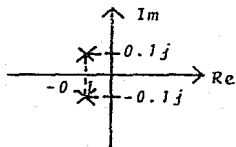


Fig. A.1.22 Patrón de polos y ceros

Mediante el siguiente desarrollo,

$$Y(S) = \frac{0.01}{S[(S+0.1)^2 + 0.01]} = \frac{A}{S} + \frac{B + CS}{(S+0.1)^2 + 0.01}$$

$$A = 0.5 \qquad B = -0.1 \qquad C = -0.5$$

$$Y(S) = \frac{0.5}{S} - \frac{0.1}{(S+0.1)^2 + 0.01} - \frac{0.5S}{(S+0.1)^2 + 0.01}$$

llegando a que la respuesta escalón es,

$$y(t) = 0.5u_{-1}(t) - 10e^{-.1t} \text{sen } 0.1t - 0.5e^{-.1t} \text{cos } 0.1t$$

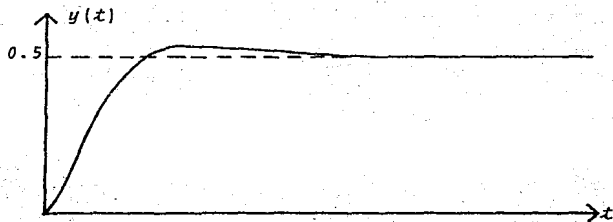
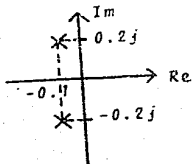


Fig. A.1.23. Respuesta escalón

Cuando $K=4$

$$H(S) = \frac{0.01}{S^2 + 0.2S + 0.05}$$

Los valores de ξ y ω_n son 0.4472 y $\sqrt{0.05}$ respectivamente, por tanto es un sistema del mismo tipo que el anterior.



A.1.24 Patrón de polos y ceros

$$Y(S) = \frac{0.01}{S((S+0.1)^2 + 0.04)} = \frac{A}{S} + \frac{B + CS}{(S+0.1)^2 + 0.04}$$

$$A = 0.2 \qquad B = -0.04 \qquad C = -0.2$$

$$Y(S) = \frac{0.2}{S} - \frac{0.04}{(S+0.1)^2 + 0.04} - \frac{0.2S}{(S+0.1)^2 + 0.04}$$

La respuesta escalón en este caso es,

$$y(t) = 0.2u_{-1}(t) - e^{-.1t} \text{sen } 0.2t - e^{-.1t} \text{cos } 0.2t$$

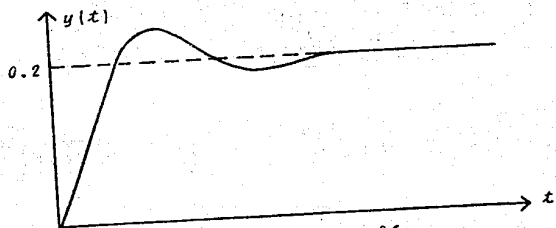


Fig. A.1.25 Respuesta escalón

A.1.7. Implementación

Material

- 5 circuitos integrados LS741
- 1 circuito integrado TL083
- 5 potenciómetros de 10 K Ω
- 5 perillas
- 12 interruptores
- 4 capacitores de 10 μ F no polarizados
- 2 capacitores de 1 μ F no polarizados
- 13 resistencias de 1 M Ω
- 1 resistencia de 270 K Ω
- 18 bornes

El circuito LS741 se utilizó únicamente para implementar el inversor y el filtro ya que funcionando como integrador, antes de aplicar la entrada escalón, ya había integrado el offset. Por tanto, se empleó el circuito TL083, que es un amplificador ya compensado. Sin embargo, con este circuito, la señal a la salida aún tenía un pequeño offset, lo que se corrigió conectando una resistencia (6.8M Ω) en paralelo con el capacitor.

El circuito de dos de los amplificadores conectados en serie (A3 y A4) está realimentado como se muestra en la figura.

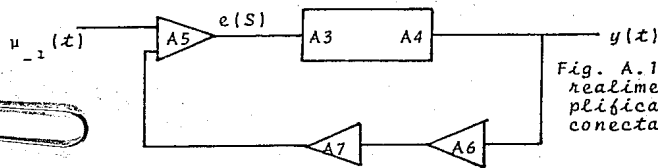


Fig. A.1.26. Circuito - realimentado de los am - plificadores A3 y A4 - conectados en cascada.

La salida del circuito A5 debe ser $e(t) = u_{-1}(t) - Ky(t)$. Por lo que debido a que los amplificadores funcionan como inversores, a la entrada, el escalón debe ser negativo y el producto $Ky(t)$, positivo. Para que $Ky(t)$ sea positivo, se introduce un inversor (A7).

A.1.8. Gráficas

Para mostrar el funcionamiento del simulador se anexan las curvas de la respuesta escalón de las siguientes funciones de transferencia.

PRIMER ORDEN

$$G(S) = \frac{2}{TOS + 1} \quad y$$

$$G(S) = \frac{1}{TOS + 1}$$

Fig. A.1.27

SEGUNDO ORDEN

$$G(S) = \frac{0.5}{(TOS+1)(TOS+1)} \quad y$$

$$G(S) = \frac{1}{(TOS+1)(TOS+1)}$$

Fig. A.1.28

$$G(S) = \frac{1}{(S+1)(S+1)} \quad y$$

$$G(S) = \frac{2}{(S+1)(S+1)}$$

Fig. A.1.29

SEGUNDO ORDEN CON REALIMENTACION

$$G(S) = \frac{1}{(TOS+1)(TOS+1)}$$

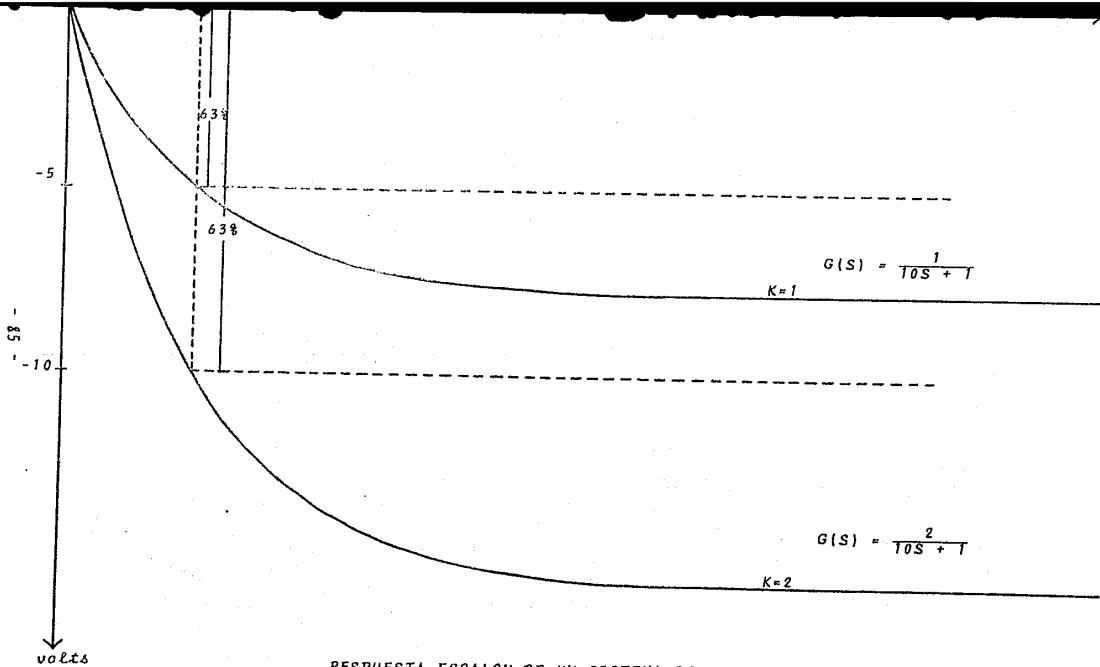
Si $K=1$

$$H(S) = \frac{0.01}{S^2 + 0.2S + 0.02}$$

Si $K=4$

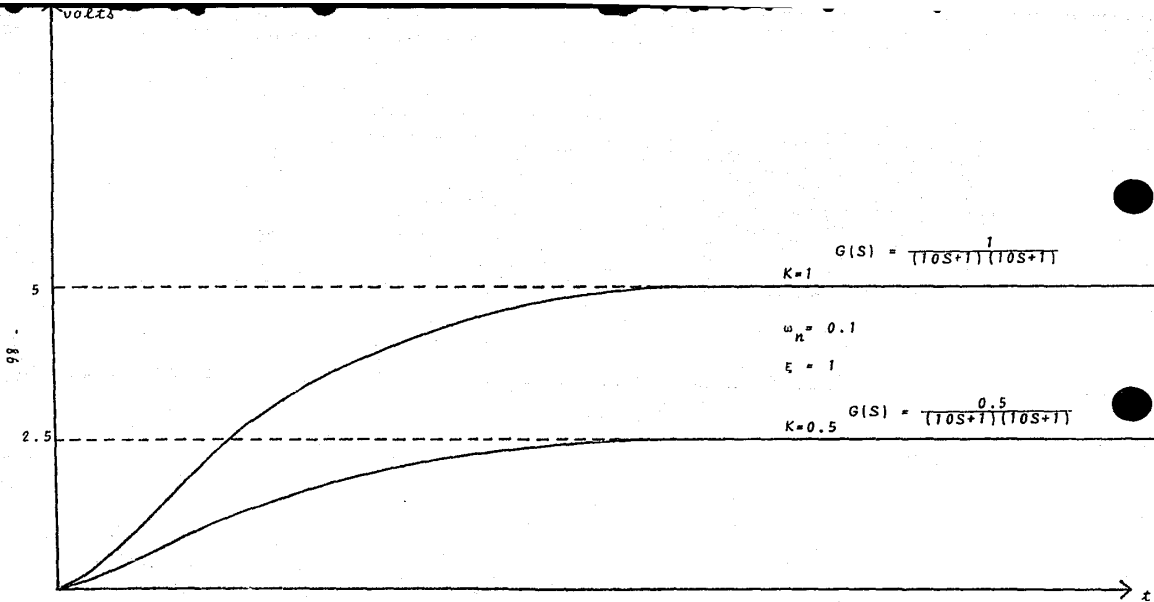
$$H(S) = \frac{0.01}{S^2 + 0.2S + 0.05}$$

Fig. A.1.30



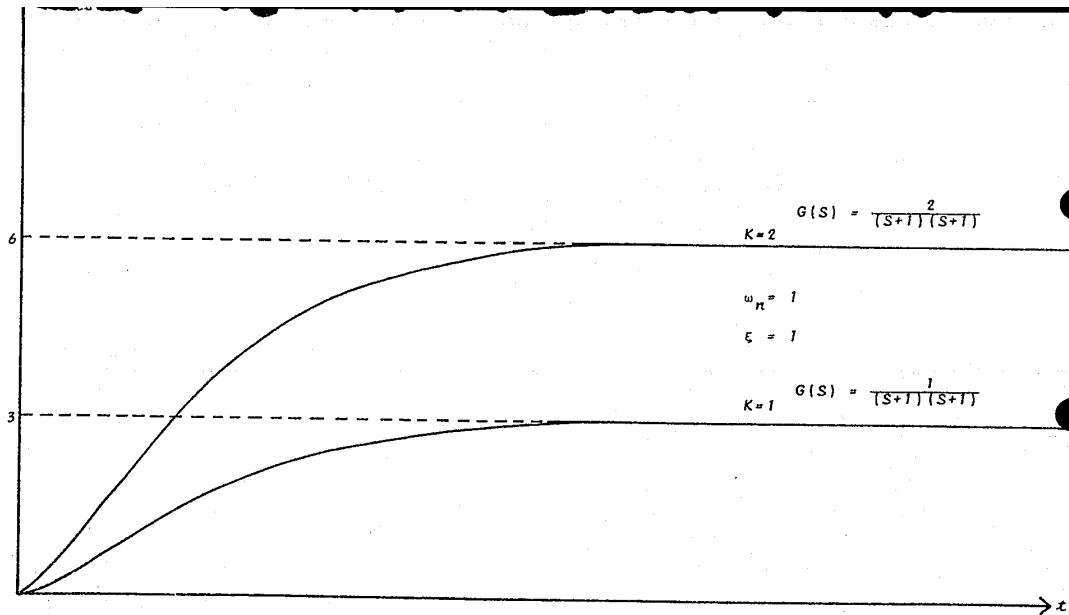
RESPUESTA ESCALON DE UN SISTEMA DE PRIMER ORDEN

FIG. A.1.27



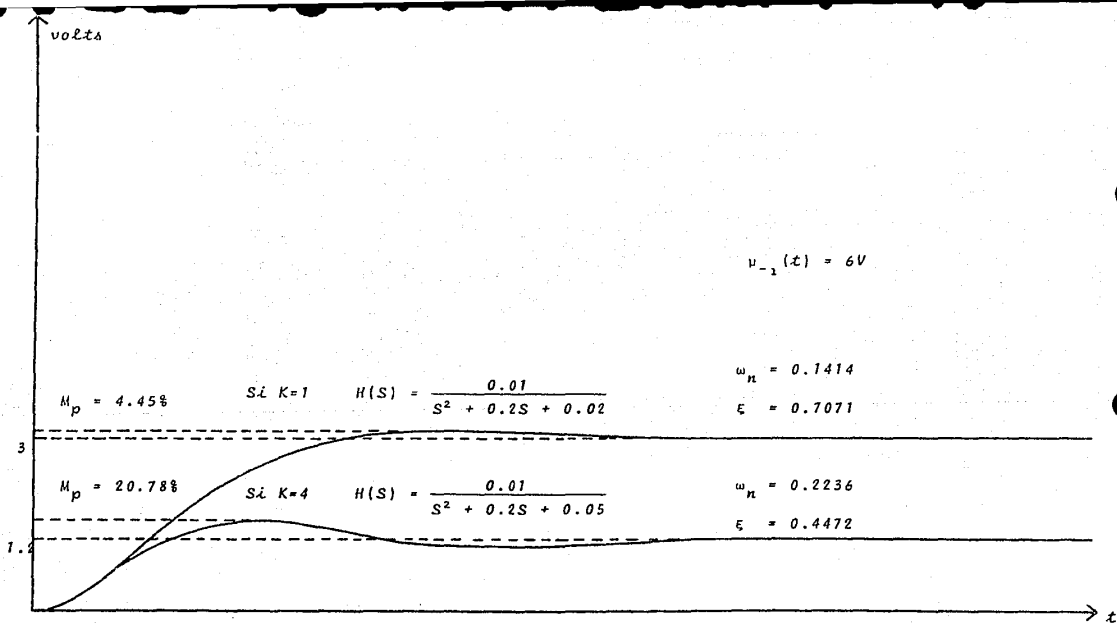
RESPUESTA ESCALON DE UN SISTEMA DE SEGUNDO ORDEN

FIG. A.1.28



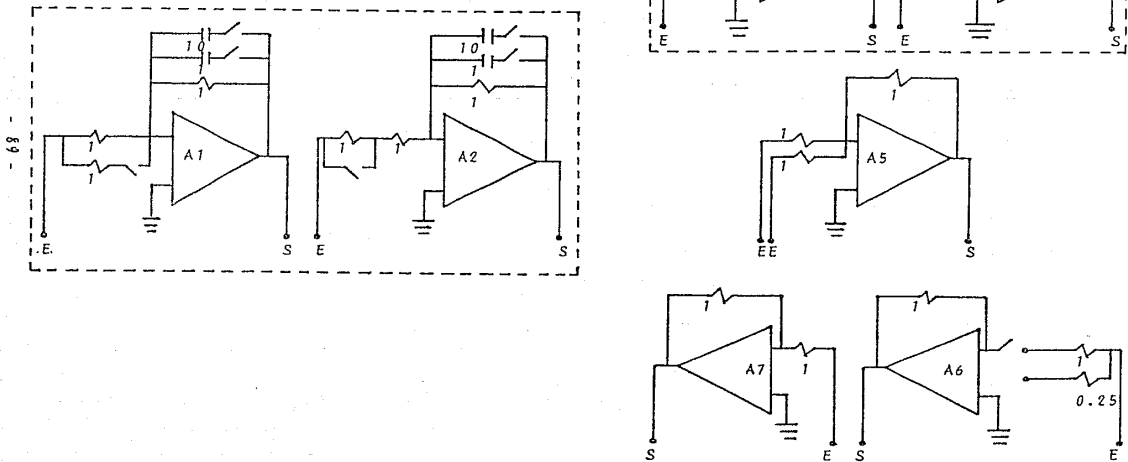
RESPUESTA ESCALON DE UN SISTEMA DE SEGUNDO ORDEN

FIG. A.1.29



RESPUESTA ESCALÓN DE UN SISTEMA DE SEGUNDO ORDEN

FIG. A.1.30



Fíg. A. 1.31

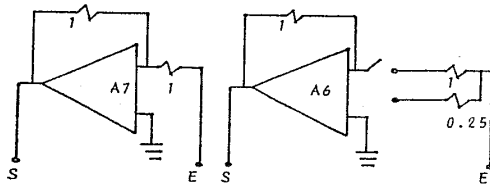
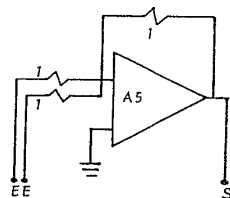
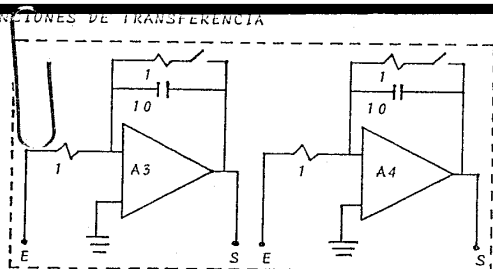
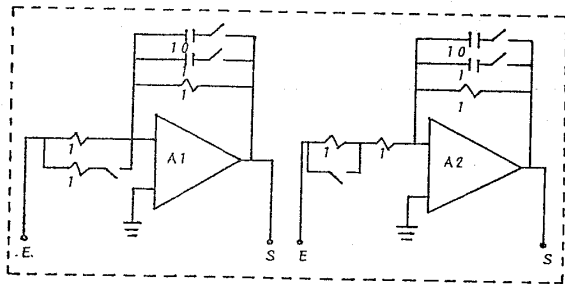
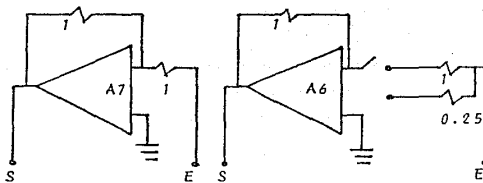
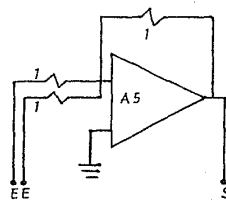
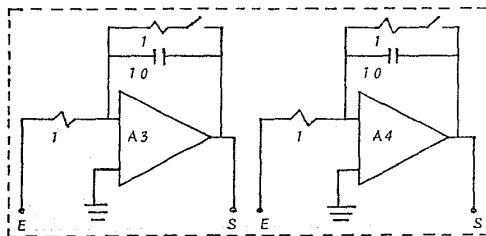
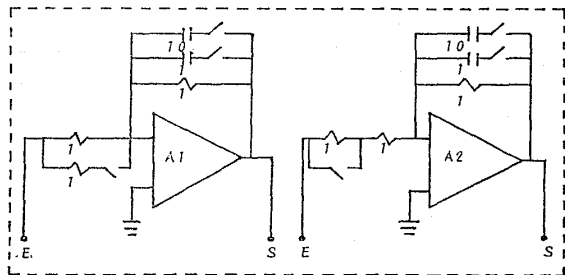
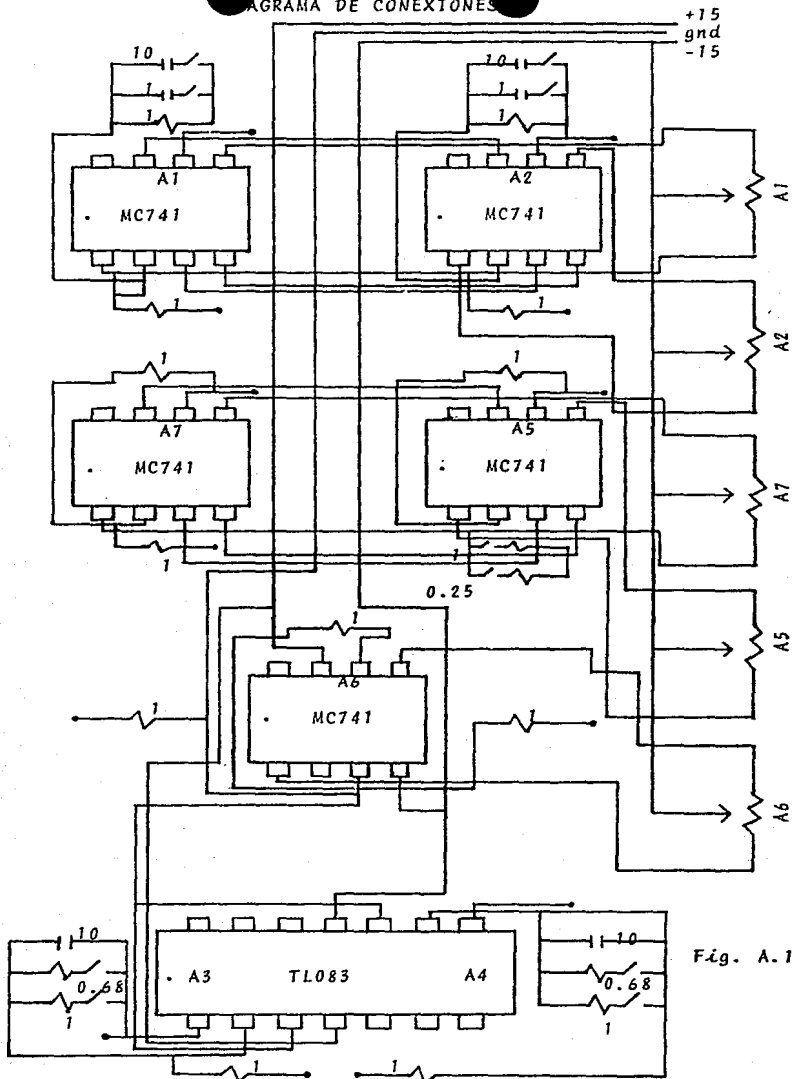


Fig. A.1.31



F4g. A.1.31

DIAGRAMA DE CONEXIONES



POTENCIOMETROS DE AJUSTE DE OFFSET

Fig. A.1.32

A.2.- SISTEMA SERVOMODULAR MS150

A.2.1.- Descripción general de un motor de C.D.

El motor de C.D. es una máquina eléctrica que transforma la energía eléctrica de C.D. en energía mecánica rotatoria.

Básicamente está formado por un rotor que contiene el devanado inducido y un estator en donde se instala el devanado inductor.

El rotor o armadura se construye de láminas de material ferromagnético (hierro o acero) aisladas una de otra por una capa de barniz o de papel gino.

El estator está formado por una carcasa construida en muchas ocasiones de una sola pieza, sobre la que se colocan los núcleos polares. Estos últimos son laminados como la armadura y a su alrededor se instalan las bobinas inductoras.

Sobre el eje de la armadura se instala el colector, Este está compuesto por delgas de cobre, aisladas entre sí, sobre las cuales se soldan las terminales del devanado de armadura. La misión del colector es dar paso a la corriente que alimenta al inducido, desde las escobillas que lo conectan al circuito exterior.

Las escobillas construidas de cobre o de carbón, se instalan en el portaescobillas, cuyo objeto es soportarlas y mantenerlas en la posición adecuada sobre el colector. Los portaescobillas se montan a su vez en un puente, que permite cambiar la posición de las escobillas, adaptándose a las condiciones de operación del motor.[3].

A.2.2.- Elementos del Sistema Servomodular MS150.[4].

Está constituido por nueve módulos:

OUI50A - Unidad Operacional

AUI50B - Unidad Atenuadora

PA150C - Unidad Preamplificadora

SA150D - Servoamplificador

PS150E - Fuente de poder

MT150F - Unidad Motor

IP150H - Potenciómetro de entrada

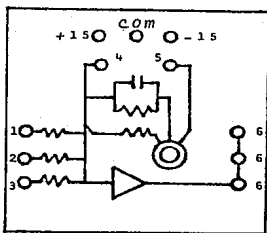
OP150K - Potenciómetro de salida

LU150L - Unidad de carga (freno magnético)

Unidad Operacional

Es un amplificador operacional conectado como sumador inversor con tres entradas. Cuenta con un ajuste a cero del offset. Opera con +15 y -15 volts. Tiene tres formas de realimentación por medio de un selector: resistencia, resistencia-capacitor y externa.

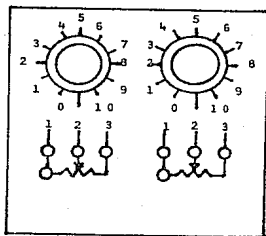
Fig. A.2.1 Unidad Operacional



Unidad Atenuadora

Contiene dos potenciómetros de 10K que se utilizan para fijar un voltaje de referencia cuando están conectados a una fuente de corriente directa o como control de ganancia cuando están conectados a la salida de un amplificador. La proporción de las resistencias está graduada en escala de 1 a 10, indicada por una perilla.

Fig. A.2.2 Unidad atenuadora



Unidad Preamplificadora

Conectando su salida 3 y 4 a cada uno de los transistores de potencia de la unidad SA150D (servoamplificador) da la bidireccionalidad al motor.

Dependiendo del signo de la suma de las entradas 1 y 2 se tiene un voltaje, si es positivo, en la salida 3 y si es negativo, en la salida 4.

Su gráfica de entrada/salida es,

Fig. A.2.3. Gráfica de entrada/salida de la unidad preamplificadora

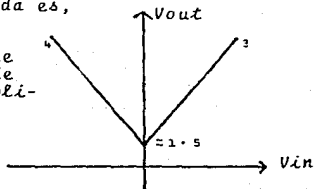
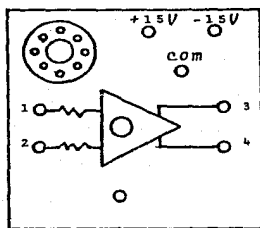


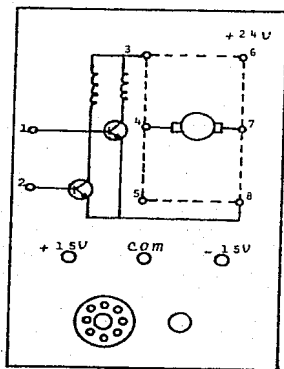
Fig. A.2.4 Unidad preamplificadora



Servoamplificador

Contiene los transistores que manejan al motor en cualquier dirección. La entrada 1 hace que el motor gire en un sentido y la entrada 2 en el opuesto. Las terminales 3,4,5,6,7 y 8 pueden interconectarse para obtener los dos modos de control, armadura y campo.

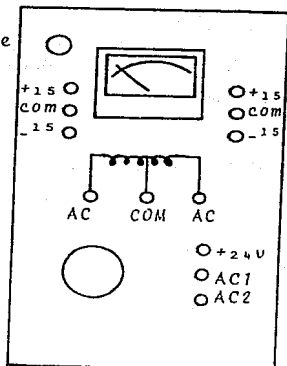
Fig. A.2.5. Servo-
amplificador



Fuente de poder

Provee 24 Vcd no regulados a 2 amp. que son suministrados al motor. -
Para alimentar a los amplificadores así como para obtener voltajes de re-
ferencia se tienen fuentes de +15 y -15V. También consta de un medidor de
corriente con indicador de sobrecarga a 2 amp. y protección.

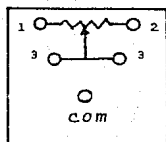
Fig. A.2.6. Fuente
de poder



Potenciómetros de entrada y salida

Son potenciómetros rotatorios de 10K; se utilizan para el control de posición. Tienen un radio de giro de $\pm 150^\circ$.

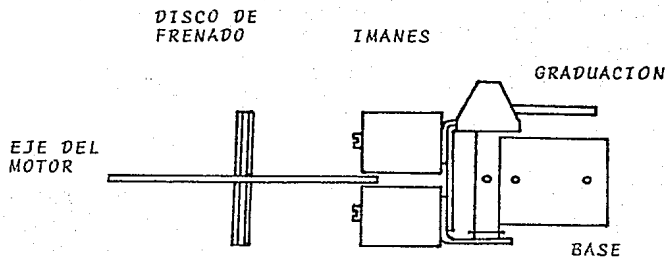
Fig. A.2.7. Potenciómetros de entrada/salida



Unidad de carga

Cuando el disco de aluminio que está montado en el eje del motor gira entre los polos del imán de la unidad de carga, las corrientes de EDDY hacen el efecto de un freno. La potencia del freno magnético puede ser controlada por la posición del imán en el disco de aluminio llamado también disco de freno.

Fig. A.2.8. Unidad de carga



Motor de Corriente Directa

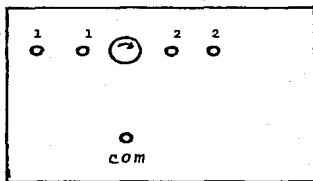
Esta unidad está formada por tres partes: motor de C.D., generador y eje de baja velocidad.

El motor de C.D. tiene campo independiente y devanado en serie con una extensión del eje para montar el disco de freno.

El generador, con salida en la parte superior, nos da en voltaje la velocidad del motor.

El eje de baja velocidad es perpendicular al eje del motor y se utiliza para el control de posición; un acoplador nos permite conectarlo al potenciómetro de salida. Está controlado por el eje de alta velocidad a través de una reducción de engranes en una relación 30:1.

Fig. A.2.9. Generador



A.3.- DESCRIPCIÓN GENERAL DEL MÓDULO DE EVALUACIÓN DEL TMS32010 (EVM).

El EVM consta de dos microprocesadores conectados en una relación maestro-esclavo: el TMS9995 y el TMS32010, respectivamente. El primero es usado para manejar el sistema operativo de la tarjeta: cargar programas, modificarlos, comandar su ejecución, etc. El segundo, es un coprocesador aritmético del primero, encargado de ejecutar los programas de aplicación.

Se tienen varias velocidades de transmisión: 110, 300, 600, 1200, 4800, 9600 y 19200 bauds, seleccionables por el comando del monitor BAUD.- La velocidad del puerto 1 (terminal del usuario) trabaja en general a 9600 bauds. El EVM se comunica al exterior por medio de tres puertos.

- Puerto 1. Terminal del usuario
- Puerto 2. Impresora o CPU huésped
- Puerto 3. Grabadora

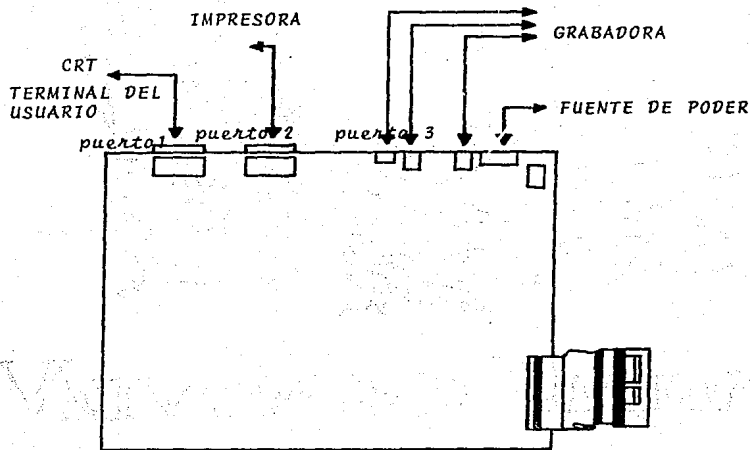


Fig. A.3.1. Módulo de evaluación del TMS32010 (EVM)

A.3.1. Memoria

La memoria del EVM está dividida en tres partes principales: EPROM - sistema operativo, RAM del sistema operativo y RAM del usuario. El mapa de memoria correspondiente se muestra en la figura siguiente:

RAM INTERNA DEL TMS9995	F000-F0FB
RAM PARA BREAKPOINT	C000-DFFF
RAM DEL USUARIO	A000-BFFF
RAM DEL SISTEMA	8000-9FFF
FIRMWARE (EPROM)	6000-7FFF
FIRMWARE (EPROM)	4000-5FFF
FIRMWARE (EPROM)	2000-3FFF
FIRMWARE (EPROM)	0000-1FFF

Los programas contenidos en la EPROM del sistema operativo se dividen en tres partes principales: el monitor, el ensamblador y el editor de texto. Cada uno de ellos realizando las funciones clásicas respectivas.

A.4.- DESCRIPCIÓN GENERAL DEL MICROPROCESADOR TMS32010

El TMS32010 es un microprocesador de señales digitales diseñado para soportar un amplio rango de aplicaciones de alta velocidad o grandes cantidades de cálculos numéricos (análisis de voz, procesamiento de imágenes, control digital, etc.).

Sus principales características son:

- + Ciclo de instrucción de 200 nS.
- + RAM de 288 bytes para datos dentro del circuito
- + Palabra de instrucción y datos de 16 bits
- + Acumulador de 32 bits
- + Multiplicación de 16 X 16 bits en 200 nS.
- + 8 puertos de entrada/salida de 16 bits cada uno
- + Bus de datos bidireccional de 16 bits con ancho de banda de 40 Mbits por seg.
- + Aritmética de punto fijo en complemento a 2.
- + Fuente de alimentación de +5 Volts
- + Encapsulado de 40 patas

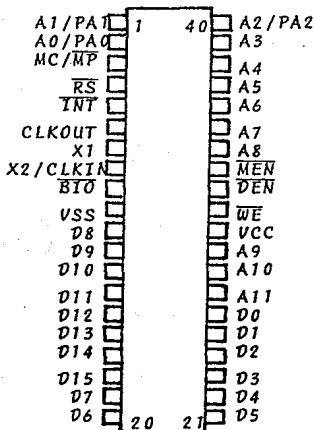


Fig. A.4.1. MICROPROCESADOR TMS32010

A.4.1.- Arquitectura

El procesador TMS32010 posee una arquitectura que permite la comunicación entre el espacio de memoria de programa y el espacio de memoria de datos (llamada Harvard modificada).

De acuerdo al diagrama de bloques de su arquitectura, los elementos básicos que lo constituyen son los siguientes: elementos aritméticos, memoria de programa, memoria de datos y registros.

A.4.1.1.- Elementos aritméticos

Los elementos aritméticos son: la ALU, el acumulador, el multiplicador y registros de corrimiento. Todas las operaciones se realizan en complemento a 2.

ALU. Es la unidad aritmética y lógica de propósito general que opera con palabras de 32 bits.

Acumulador. El acumulador es siempre el destino de la operación y contiene el operando primario. Almacena la salida de la ALU y también proporciona un dato a ésta.

Multiplicador de 16 X 16 bits. Consta de tres unidades: el registro T, el registro P y el multiplicador. El registro T es un registro de 16 bits que almacena el multiplicando; el multiplicador viene de la memoria de datos para la instrucción MPV, o en el caso de una multiplicación inmediata (MPYK), forma parte del código de la instrucción.

Registros de corrimiento. Hay dos registros de corrimiento para manipular los datos; uno de ellos es para corrimiento de datos que van de la RAM de datos a la ALU. Realiza un corrimiento a la izquierda de 0 a 15 posiciones en todas las palabras de datos que serán cargadas, restadas o sustradas al acumulador. El otro es el registro de corrimiento paralelo; se activa solamente mediante la instrucción SACH; este registro solo ejecuta corrimientos de 0, 1 ó 4 posiciones a la izquierda.

A.4.1.2. Registros internos

Registros auxiliares (ARO y AR1). Son registros de 16 bits, que no es tán contenidos en la RAM de datos. Se utilizan como almacenamiento tempo-ral, para llevar a cabo el direccionamiento indirecto de la memoria de datos y control de lazos (loops).

Registro apuntador auxiliar. Es parte del registro de estados e indi-ca cual de los registros auxiliares se está utilizando.

Apuntador de programa (PC). Es un registro de 12 bits que contiene la dirección de la siguiente instrucción a ejecutar. Se inicializa en cero al activar la línea de RESET. Los tres bits menos significativos del PC son utilizados para direccionar los puertos de entrada/salida.

Pila (Stack). Está formado de 4 localidades de 12 bits cada una; para escribir o leer de la pila se utilizan las instrucciones PUSH y POP respectivamente, que operan con los 12 bits menos significativos del acumulador.

El PC y el Stack permiten al usuario realizar saltos, llamadas a sub-rutinas, interrupciones y ejecutar las instrucciones TBIR y TBLW.

Registro de estados. Este registro está formado por 5 bits donde almaena 5 banderas. Los bits pueden ser alterados individualmente mediante -instrucciones dedicadas. Además el registro puede ser almacenado en memo -ria de datos mediante la instrucción SST. Las banderas correspondientes -son las siguientes:

OV bandera de desbordamiento del acumulador

OVM bit de modo desbordamiento

INTM bit de enmascaramiento de interrupciones

ARP apuntador de registro auxiliar

DP apuntador de página de memoria de datos

A.4.1.3 Buses

A.4.1.3.1.- Buses internos

Bus de programa. Tiene 16 líneas, su función es la de interconectar -

la memoria de programa con el PC, la pila y el resto de los componentes -- del circuito integrado.

Bus de datos. Tiene 16 líneas también y comunica a la memoria de datos con los elementos aritméticos y el resto del circuito integrado.

Existe una interconexión entre estos dos buses que permite el flujo - de información entre las dos memorias.

A.4.1.3.2.- Buses externos

Bus de datos. Es un bus bidireccional de 16 bits con una velocidad máxima de transmisión de 40 Mbits por seg.

Bus de direcciones. Es un bus unidireccional de 12 bits, que permite direccionar la memoria y los puertos de entrada/salida.

A.4.1.4.- Funciones de entrada/salida

La entrada y salida de datos de un periférico se lleva a cabo por las instrucciones IN y OUT respectivamente, a través de uno de los 8 puertos - direccionables desde el microprocesador, como se muestra en la fig. A.4.2. Los datos se transfieren por el bus de datos de 16 bits, a y de, la memoria de datos; la transferencia está sincronizada con las señales independientes: habilitación de datos (\overline{DEN}) (para una lectura, IN) y habilitación de escritura (\overline{WE}) (para la instrucción OUT).

El bus de datos externo bidireccional se encuentra siempre en estado de alta impedancia, excepto cuando alguna de esas dos líneas de control están en su estado activo bajo. Como se muestra en las figs. A.4.3 y A.4.4., \overline{WE} estará en su estado bajo durante una instrucción OUT y la señal \overline{DEN} durante una instrucción IN.

Para direccionar los puertos se utilizan los 3 bits menos significativos del bus de direcciones, (PA2 a PA0). Los 9 bits restantes de este bus se mantienen en su nivel bajo durante la ejecución de estas instrucciones.

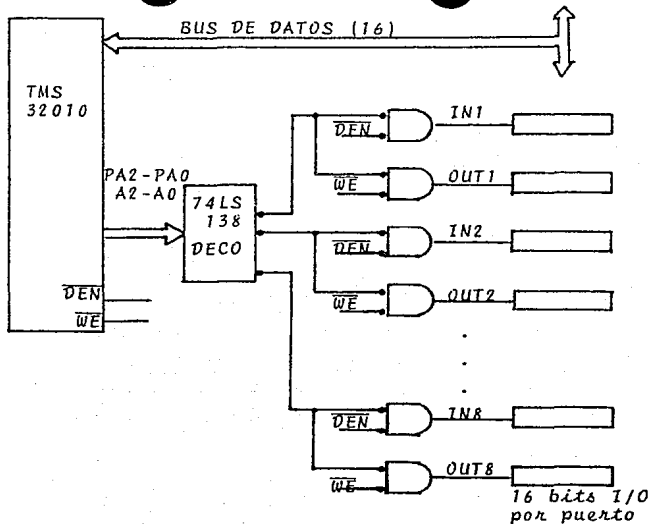


Fig. A.4.2. Direcccionamiento de los puertos I/O.

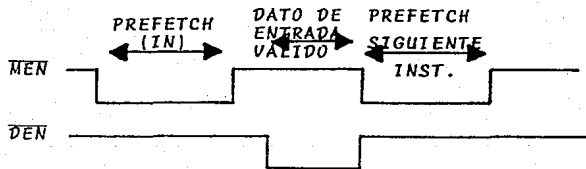


Fig. A.4.3. Diagramas de tiempo de la instrucción IN

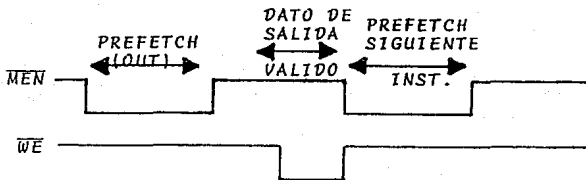


Fig. A.4.4 Diagramas de tiempo de la instrucción OUT

A.4.1.5. - Memoria

A.4.1.5.1. - Memoria de programa

Dependiendo de la versión del microprocesador hay dos modos de operación definidos por el estado de la entrada MC/\overline{MP} : el modo microcomputador y el modo microprocesador.

La tabla siguiente ilustra el tamaño y la distribución de la memoria de programa de la familia TMS320, para cada uno de los modos de operación.

VERSION	Opciones de la memoria de programa	Modo microcomputador	Modo microprocesador
TMS320M10	Modos microcomputador y microprocesador.	$MC/\overline{MP} = 1$ 1536 palabras dentro del circuito (ROM) y 2560 palabras de memoria externa.	$MC/\overline{MP} = 0$ 4096 palabras de memoria externa.
TMS32010	Modo microprocesador solamente	No utilizado	4096 palabras de memoria externa

Como se observa la versión TMS32010, que es la que se utilizó en esta tesis, solo soporta el modo microprocesador en donde los 4K palabras de la memoria de programa están externas al circuito. Para configurar esta memoria se aconseja utilizar memoria RAM estática con un tiempo de acceso inferior a 100 nS., para no penalizar en tiempo al microprocesador.

El bus de direcciones de 12 líneas se utiliza para el direccionamiento de la memoria externa. Los bits de la dirección están contenidos en el contador de programa. Cuando una instrucción está en el ciclo de FETCH, la señal \overline{MEN} será generada para habilitar la memoria externa. Se hace notar que \overline{MEN} nunca se activa al mismo tiempo que las señales \overline{WE} o \overline{DEN} . En efecto, \overline{MEN} será baja en cada ciclo de reloj excepto cuando una función I/O se ejecuta por las instrucciones IN o OUT respectivamente.

En instrucciones de varios ciclos, \overline{MEN} es baja durante los ciclos en

que \overline{WE} y \overline{DEN} no están activas.

A.4.1.5.2.- Memoria de datos

El microprocesador cuenta con una memoria de datos de 144 palabras de 16 bits en una RAM dentro del circuito.

Por el tamaño de esta RAM algunas veces es necesario guardar operandos fuera del circuito y leerlos cuando se necesitan. Para ello existen las instrucciones TBLR y TBLW.

La instrucción lectura de tabla (TBLR) transfiere valores de la memoria de programa a la memoria de datos.

La instrucción de escritura de tabla (TBLW) transfiere valores de la RAM de datos a la memoria de programa. Hay que hacer notar que estas instrucciones se ejecutan en 3 ciclos de reloj del microprocesador, por lo que siempre que sea posible es preferible utilizar las instrucciones IN, OUT, que se ejecutan en solo dos ciclos.

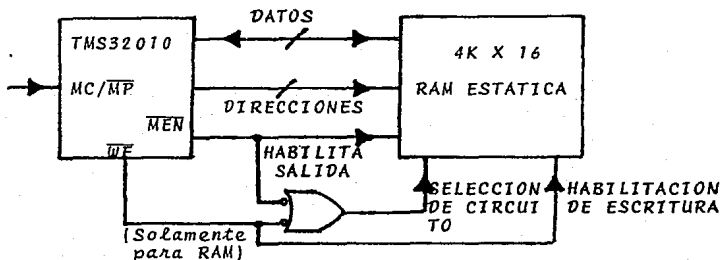


Fig. A.4.5. Memoria de programa (externa al TMS32010)

A.4.2 Interrupciones

Existen dos formas de interrumpir al microprocesador:

a) Aplicando un voltaje bajo en la entrada \overline{BIO} .

Cuando un voltaje bajo se presenta en la entrada \overline{BIO} , la ejecución de la instrucción $BIOZ$ causará la ejecución de una subrutina preparada para este caso.

La entrada \overline{BIO} es usada para monitorear el estado de un periférico. Se utiliza especialmente como una alternativa para interrumpir cuando no se requiere que el microprocesador realice otras instrucciones mientras se espera que el voltaje en la entrada \overline{BIO} sea bajo.

b) Aplicando un voltaje bajo en la entrada \overline{INT}

La interrupción en sentido estricto se genera aplicando un pulso negativo o manteniendo baja la entrada \overline{INT} . El diagrama del circuito de interrupción interno del TMS32010 se presenta en la siguiente figura.

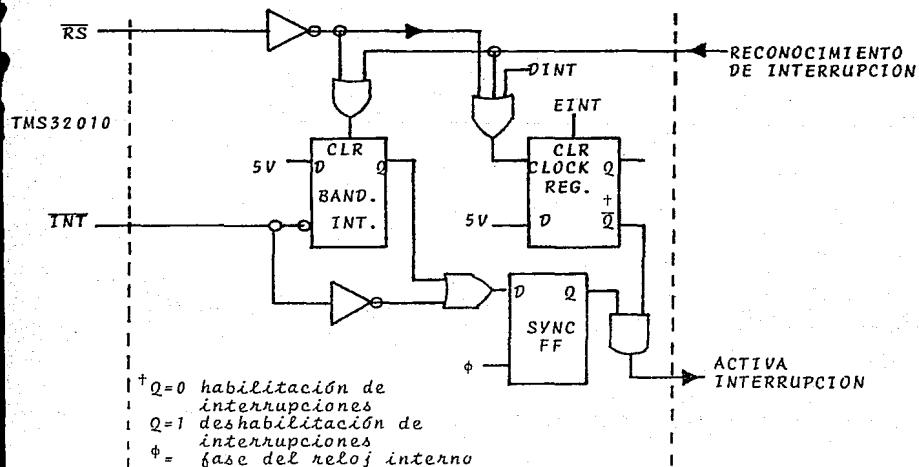


Fig. A.4.6. Diagrama del circuito de interrupción interno del TMS32010

Como se puede observar el FLIP-FLOP SYNC FF se utiliza para sincronizar la señal de interrupción externa, al circuito de interrupción interno.

Si el registro de modo de interrupción, que depende de la máscara de interrupción (INTM) es 1, entonces una señal externa de interrupción causará un salto a la localidad 2 en la memoria de programa. En dicha localidad se indica el inicio de la rutina de servicio de la interrupción.

La solicitud de interrupción esperará en cada uno de los siguientes casos:

1. Hasta finalizar todos los ciclos de una instrucción con varios ciclos.

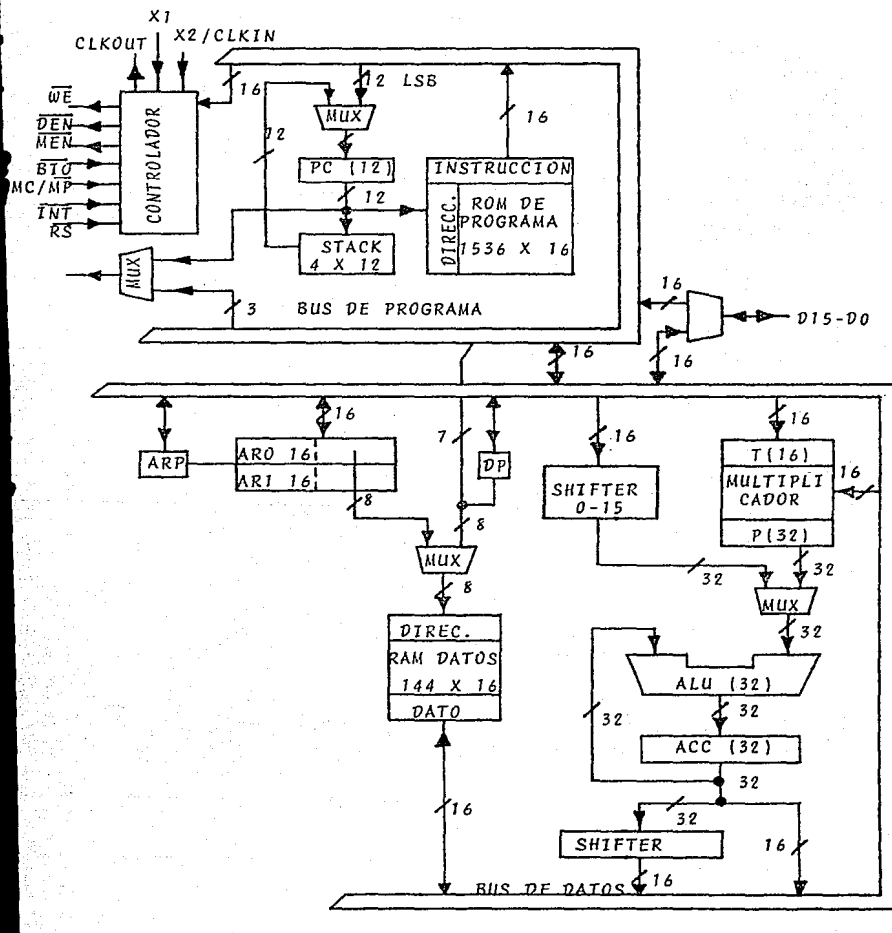
2. Hasta que se haya completado la ejecución de una multiplicación - MPY ó MPYK.

3. Cuando las interrupciones han sido deshabilitadas previamente, es necesario ejecutar la instrucción EINT que las habilita nuevamente.

Este último punto permite que se pueda regresar de la subrutina de interrupción, con la instrucción RET, antes de la llegada de una nueva interrupción ya que éstas quedan deshabilitadas después de la aceptación de una de ellas.

Fig. A.4.7.

DIAGRAMA DE BLOQUES DE LA ARQUITECTURA DEL TMS32010



A.5.- DESCRIPCION DE LA TARJETA DE INTERFAZ ANALOGICA

La tarjeta de interfaz analógica es usada como una tarjeta complementaria del módulo de evaluación TMS32010.

Esta tarjeta está constituida por:

- + Un convertidor A/D de 12 bits
- + Un convertidor D/A de 12 bits
- + Un puerto de salida de 16 bits para un convertidor D/A adicional o una aplicación definida por el usuario
- + Un puerto de entrada de 16 bits para un convertidor A/D adicional o una aplicación definida por el usuario
- + Dos filtros pasabajas
- + Un decodificador TBLW
- + Memoria de expansión, de datos
- + Registro de control

El diagrama de bloques siguiente muestra la arquitectura básica.

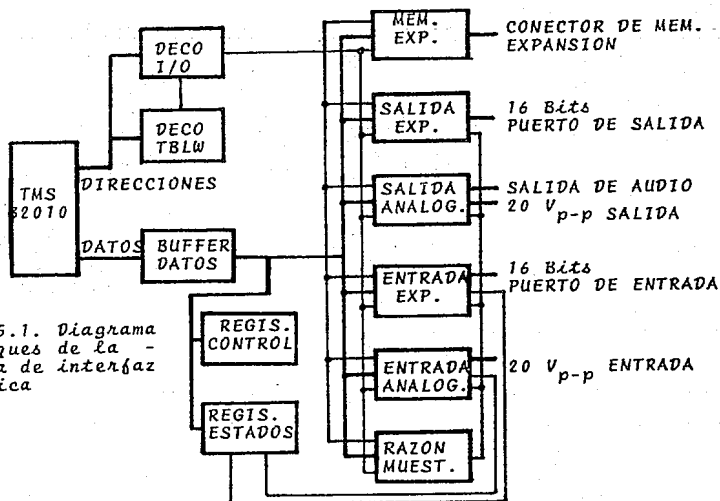


Fig. A.5.1. Diagrama de bloques de la tarjeta de interfaz analógica

A.5.1 Definiciones de los puertos de entrada/salida

Los datos son transferidos entre el módulo de evaluación y la tarjeta de interfaz analógica con las instrucciones IN y OUT descritas en A.4.1.4.- La siguiente tabla indica la función de cada uno de los puertos asociados a estas operaciones de entrada/salida.

Puerto	Función de entrada	Función de salida
0	Lee el registro de estados del convertidor A/D	Carga el registro de control de la tarjeta AIB.
1	No se usa	Carga la razón de muestreo
2	Lee los datos del convertidor A/D	Escribe los datos en el convertidor D/A
3	Puerto de expansión (lectura)	Puerto de expansión (escritura)
4	Lectura de la dirección inicial de la memoria de expansión	Carga la dirección inicial de la memoria de expansión
5	Lectura de datos de la memoria de expansión	Escritura de datos en la memoria de expansión
6	No se usa	No se usa
7	No se usa	No se usa

A.5.2. Registro de control de la tarjeta de interfaz analógica

El registro de control es cargado por el TMS32010 para definir el modo de operación de la tarjeta. Un modo específico puede ser seleccionado, guardado en la memoria de datos y enviado al puerto 0 con la instrucción -- OUT.

El registro de control es definido de la siguiente manera:

15	8	7	6	5	4	3	2	1	0
No se usa		DCW	DCR	U/D	CAD2	CAD1	CDA2	CDA1	CCLR
BIT 0 (CCLR)		1 Razón de muestreo deshabilitada		0 Razón de muestreo habilitada					
BIT 1 (CDA1)		1 Modo transparente para D/A		0 Modo de muestreo para D/A					
BIT 2 (CDA2)		1 Modo transparente para expansión D/A		0 Modo de muestreo para expansión D/A					

- BIT 3 (CAD1) 1 Modo de recepción automática para A/D
0 Modo de recepción asíncrona para A/D
- BIT 4 (CAD2) 1 Modo de recepción automática para expansión A/D
0 Modo de recepción asíncrona para expansión A/D
- BIT 5 (U/D) 1 Contador hacia arriba de la dirección de la memoria de expansión
0 Contador hacia abajo de la dirección de la memoria de expansión
- BIT 6 (DCR) 1 Contador de dirección de la memoria de expansión deshabilitado para lectura de datos (IN)
0 Contador de dirección de la memoria de expansión habilitado para lectura de datos (IN)
- BIT 7 (DCW) 1 Contador de dirección de la memoria de expansión deshabilitado para escritura de datos (OUT)
0 Contador de dirección de la memoria de expansión habilitado para escritura de datos (OUT)

A.5.3. Memoria de expansión

La tarjeta AIB tiene una memoria de expansión de 8K palabras. Las señales de control, el bus de datos y el de direcciones están conectados al conector (P2) para una expansión de 64K. Esta memoria se accesa mediante las instrucciones IN y OUT.

Su diagrama de bloques se muestra en la figura siguiente:

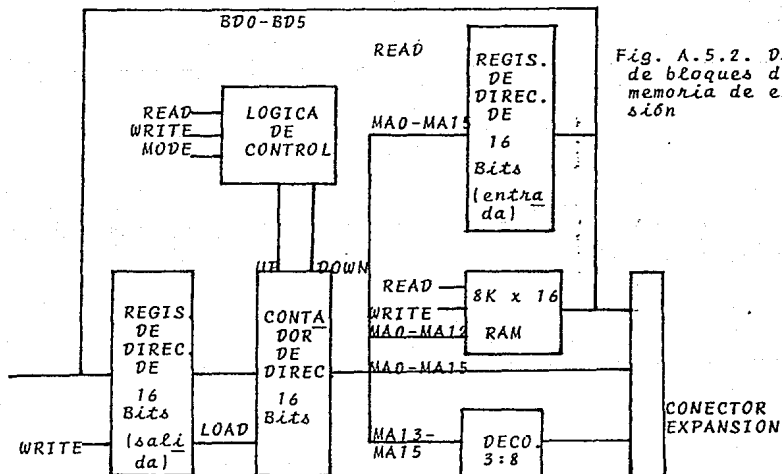


Fig. A.5.2. Diagrama de bloques de la memoria de expansión

La interfaz con la memoria consiste en un latch de dirección, un contador de direcciones, una memoria RAM estática, una memoria temporal tipo-buffer para la lectura del contador de direcciones y la lógica de control-necesaria. El contador de direcciones permite al usuario realizar autoincrementos o autodecrementos. El modo del contador es seleccionado por medio de la escritura del valor apropiado del registro de control por el puerto 0. La dirección de inicio es escrita en el latch por el puerto 4. Una escritura o una lectura en la memoria RAM de expansión se realiza por el puerto 5. Los bits de control del contador de direcciones están definidos como una parte del registro de control de la tarjeta AIB.

Los modos de direccionamiento de la memoria de expansión se proporcionan en la siguiente tabla.

DCW	DCR	U/D	MODO
0	0	0	El contador de direcciones contará hacia abajo en lectura o escritura.
0	0	1	El contador de direcciones contará hacia arriba en lectura o escritura.
0	1	0	El contador de direcciones contará hacia abajo en escritura solamente.
0	1	1	El contador de direcciones contará hacia arriba en escritura solamente.
1	0	0	El contador de direcciones contará hacia abajo en lectura solamente.
1	0	1	El contador de direcciones contará hacia arriba en lectura solamente.
1	1	0	El contador está deshabilitado
1	1	1	El contador está deshabilitado

A.5.4. Frecuencia de muestreo

El reloj que proporciona la razón de muestreo es un contador divisor por N, programable. La señal CLKOUT del TMS32010 es usada como la fuente para el contador. Los pulsos de salida del reloj inician la conversión A/D así como la generación de otras señales que controlan la salida de datos -

para el convertidor D/A. La relación de este reloj con la frecuencia de --
muestreo está dada por:

$$F_{sr} = \frac{F_{clkout}}{N + 1}$$

donde,

F_{clkout} = 5 Mhz. para el TMS32010

N = constante de 16 bits que se carga en el registro del reloj

Para inicializar esta frecuencia se manda el valor correspondiente a-
N de una localidad de memoria de datos al puerto 1 con una instrucción OUT.

El rango de frecuencias de salida es de 76.29 Hz. a 5 Mhz.

A.5.5. Filtros analógicos de la tarjeta

La ATB contiene dos filtros pasabajas:

1) Filtro limitador de banda

2) Filtro de reconstrucción

El primer filtro, limita la banda de entrada y minimiza los efectos -
de traslape en el espectro de la señal muestreada. El segundo filtro suavi-
za la salida analógica del convertidor D/A.

Estos filtros son de sexto orden de tipo Butterworth y están formados
por tres filtros de segundo orden conectados en cascada, de función de --
transferencia:

$$G(S) = \frac{-1/(R_1 R_2)}{S^2 C_1 C_2 + S C_2 [(1/R_1) + (1/R_2) + (1/R_3)] + 1/R_2 R_3}$$

A.5.6. Especificaciones generales de la tarjeta de interfaz analógica

Convertidor A/D

resolución: 12 bits

entrada analógica: -10 a 10 volts

salida digital: 16 bits en complemento a 2

tiempo de conversión: 25 uS.

Convertidor D/A

resolución: 12 bits

salida analógica: -10 a 10 volts

entrada digital: 16 bits en complemento a 2

tiempo de asentamiento: 250 nS.

Muestreo:

76.29 Hz. a 5MHz.

Memoria de datos de expansión

8192 X 16 bits

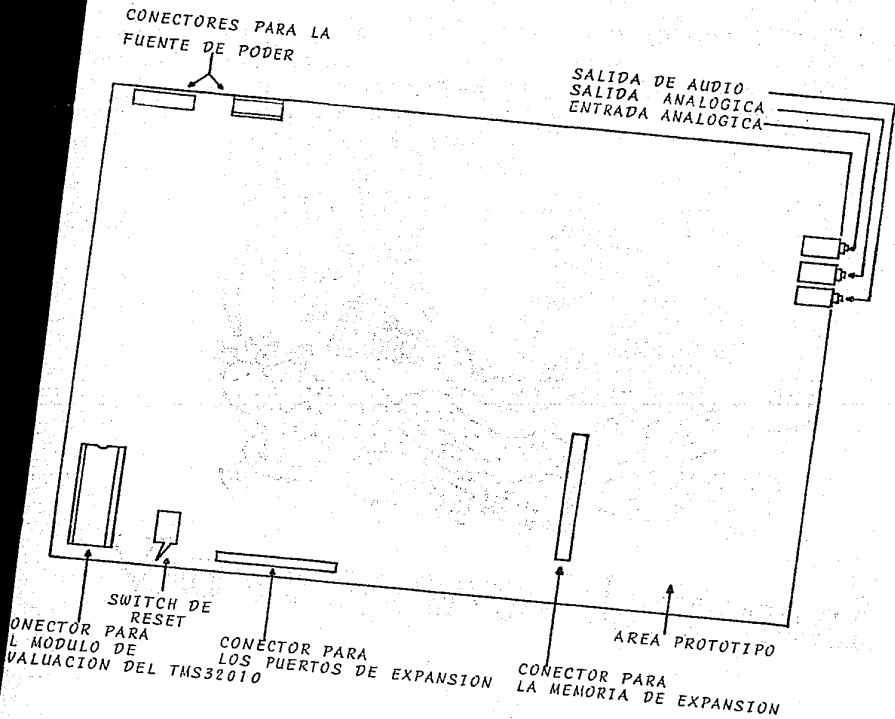
Muestreador/Retén

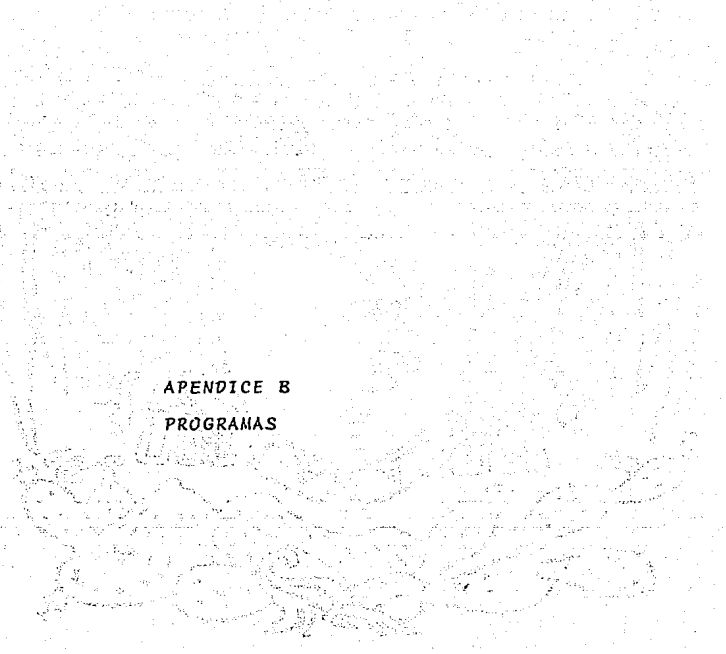
Tiempo de adquisición a 0.1%: 4 uS.

Output droop rate: 0.3 V/seg.

Hold step: 10 mV.

Fig. A.5.3.
TARJETA DE INTERFAZ ANALOGICA





APENDICE B
PROGRAMAS

UNIVERSIDAD NACIONAL

PROGRAMA 1

PROGRAMA CONTROL ADAPTABLE CON TRES PARAMETROS
(Wf SENAL CUADRADA)

*
*
*
*

B	PRIN	
KA	DATA	4096
GA	DATA	4096
TR	DATA	819
DAT1	DATA	4096
DAT2	DATA	-4096
CT	DATA	50
M3	DATA	>7FFF
ALPH	DATA	>D55
BET	DATA	>2AA
TINV	DATA	5
IPDS	DATA	4096
RATE	DATA	499
TE1	DATA	0
TE2	DATA	0
TE3	DATA	0

* LOCALIDADES UTILIZADAS EN LA MEMORIA DE DATOS

GAMA	EQU	0
TER	EQU	1
ERR	EQU	2
TEMP	EQU	3
DIF	EQU	4
TETA1	EQU	5
TETA2	EQU	6
TETA3	EQU	7
TET1	EQU	8
TET2	EQU	9
TET3	EQU	10
ALPHA	EQU	11
BETA	EQU	12
SUM1	EQU	13
SUM2	EQU	14
CDAT	EQU	15
MASK3	EQU	16
DELTA	EQU	17
ENT	EQU	18
RESUL	EQU	19
DEV	EQU	20
INV	EQU	21
MULT	EQU	22
DER	EQU	23
TERM1	EQU	24
TERM2	EQU	25
TERMS	EQU	26
I1	EQU	27
I2	EQU	28
CORR	EQU	29
VEL	EQU	30
KIN	EQU	31
CUAD	EQU	32
DIF1	EQU	33

DIF2	EQU	34
PRIN	LDPK	0
	LACK	250
	SACL	0
	OUT	0,0
	LACK	RATE
	TBLR	0
	OUT	0,1
	LACK	CT
	TBLR	CDAT
	LACK	1
	SACL	DELTA
	LACK	M3
	TBLR	MASK3
	LACK	ALPH
	TBLR	ALPHA
	LACK	BET
	TBLR	BETA
	LACK	TINV
	TBLR	INV
	ZAC	
	SACL	RESUL
	LACK	TE1
	TBLR	TETA1
	LACK	TE2
	TBLR	TETA2
	LACK	TE3
	TBLR	TETA3
	LACK	TR
	TBLR	TER
	LACK	GA
	TBLR	GAMA
	LACK	KA
	TBLR	KIN

* SEMICICLO FOBITIVO

L1	LAC	RESUL
	SACL	DEV
	LAC	TETA1
	SACL	TET1
	LAC	TETA2
	SACL	TET2
	LAC	TETA3
	SACL	TET3
	LT	BETA
	LACK	DAT1
	TBLR	ENT
	CALL	SUB3
	LAC	CDAT
	SUB	DELTA
	AND	MASK3
	SACL	CDAT
	BNZ	L1
	LACK	CT
	TBLR	CDAT

* SEMICICLO NEGATIVO

L2 LAC RESULT
SACL DEV
LAC TETA1
SACL TET1
LAC TETA2
SACL TET2
LAC TETA3
SACL TET3
LT BETA
LACK DAT2
TBLR ENT
CALL SUB3
LAC CDAT
SUB DELTA
AND MASK3
SACL CDAT
BNZ L2
LACK CT
TBLR CDAT
B L1

* SUBROUTINA DE CALCULO

**SE OBTIENE Wm

SUB3 MPY ENT
PAC
SACH SUM2,4
LT ALPHA
MPY DEV
PAC
SACH SUM1,4
LAC SUM1
ADD SUM2
SACL RESULT

**SE OBTIENE DERIVADA DE Wm

SUB DEV
SACL MULT
LT MULT
MPY INV
PAC
SACL DER

**SE LEE MUESTRA DE W

ALOOP B02 ALOOP1
B ALOOP
ALOOP1 IN VEL,2

**SE OBTIENE PRODUCTO (PERIODO) (GAMA) (W - Wm)

LAC VEL
SUB RESULT
SACL ERR

LT ERR
 MPY TER
 PAC
 SACH TEMP,4
 LT TEMP
 MPY GAMA
 PAC
 SACH DIF,4

**SE OBTIENE PRODUCTO (DIF) (DER Wm)

LT DIF
 MPY DER
 PAC
 SACH DIF1,4

**SE OBTIENE PRODUCTO (DIF) (Wm)

LT DIF
 MPY RESULT
 PAC
 SACH DIF2,4

LAC VEL
 BGEZ L8

**SI W MENOR QUE CERO

LACK 0
 SACL I1 * I1=0
 LACK IPOS
 TBLR I2 * I2=1
 LAC TET2
 SACL TETA2 * SE OBTIENE TETA2
 LAC TET3
 SUB DIF2
 SACL TETA3 * SE OBTIENE TETA3
 B L9

**SI W MAYOR O IGUAL QUE CERO

L8 LACK IPOS
 TBLR I1 * I1=1
 LACK 0
 SACL I2 * I2=0
 LAC TET2
 SUB DIF2
 SACL TETA2 * SE OBTIENE TETA2
 LAC TET3
 SACL TETA3 * SE OBTIENE TETA3

**SE OBTIENE T1=(TETA1) (DER Wm)

L9 LAC TET1
 SUB DIF1
 SACL TETA1
 LT DER

MPY TETA1
PAC
SACH TERM1,4

**SE OBTIENE T2=(I1) (TETA2) (Wm)

LT RESUL
MPY I1
PAC
SACH TEMP,4
LT TEMP
MPY TETA2
PAC
SACH TERM2,4

**SE OBTIENE T3=(I2) (TETA3) (Wm)

LT RESUL
MPY I2
PAC
SACH TEMP,4
LT TEMP
MPY TETA3
PAC
SACH TERM3,4

**SE OBTIENE I=1/K (T1+T2+T3)

LAC TERM1
ADD TERM2
ADD TERM3
SACL CORR
LT KIN
MPY CORR
PAC
SACH CUAD,4
OUT CUAD,2
CALL SUB
CALL SUB
RET

* SUBROUTINA DE RETARDO

SUB B START
CONT DATA 2
CONTA EQU 35
START LDPK 0
LACK CONT
TBLR CONTA
L4 CALL SUB1
LAC CONTA
SUB DELTA
AND MASK3
SACL CONTA
BZ L5
B L4
L5 RET

SUB1	S	INIC
CON	DATA	>7FFF
CONTD	EQU	36
INIC	LDPK	0
	LACK	CON
	TBLR	CONTD
L7	LAC	CONTD
	SUB	DELTA
	AND	MASK3
	SACL	CONTD
	RZ	L6
	B	L7
L6	RET	
	END	

PROGRAMA CONTROL ADAPTABLE CON CINCO PARAMETROS
(W/ SENAL CUADRADA)

*
*
*
*

B	PRIN
KA	DATA 4096
GA	DATA 4096
TR	DATA 819
DAT1	DATA 4096
DAT2	DATA -4096
CT	DATA 50
M3	DATA >7FFF
ALPH	DATA >D55
BET	DATA :2AA
TINV	DATA 5
IPDS	DATA 4096
RATE	DATA 499
TE1	DATA 0
TE2	DATA 0
TE3	DATA 0
TE4	DATA 0
TE5	DATA 0

* LOCALIDADES UTILIZADAS EN LA MEMORIA DE DATOS

GAMA	EQU	0
TER	EQU	1
ERR	EQU	2
TEMP	EQU	3
DIF	EQU	4
TETA1	EQU	5
TETA2	EQU	6
TETA3	EQU	7
TETA4	EQU	8
TETA5	EQU	9
TET1	EQU	10
TET2	EQU	11
TET3	EQU	12
TET4	EQU	13
TET5	EQU	14
ALPHA	EQU	15
BETA	EQU	16
SUM1	EQU	17
SUM2	EQU	18
CDAT	EQU	19
MASK3	EQU	20
DELTA	EQU	21
ENT	EQU	22
RESUL	EQU	23
DEV	EQU	24
INV	EQU	25
MULT	EQU	26
DER	EQU	27
TERM1	EQU	28
TERM2	EQU	29
TERM3	EQU	30
TERM4	EQU	31

TERMS	EQU	32
I1	EQU	33
I2	EQU	34
CORR	EQU	35
VEL	EQU	36
KIN	EQU	37
CUAD	EQU	38
DIF1	EQU	39
DIF2	EQU	40
PRIN	LDPK	0
	LACK	250
	SACL	0
	OUT	0,0
	LACK	RATE
	TBLR	0
	OUT	0,1
	LACK	CT
	TBLR	CDAT
	LACK	1
	SACL	DELTA
	LACK	M3
	TBLR	MASK3
	LACK	ALPH
	TBLR	ALPHA
	LACK	BET
	TBLR	BETA
	LACK	TINV
	TBLR	INV
	ZAC	
	SACL	RESUL
	LACK	TE1
	TBLR	TETA1
	LACK	TE2
	TBLR	TETA2
	LACK	TE3
	TBLR	TETA3
	LACK	TE4
	TBLR	TETA4
	LACK	TE5
	TBLR	TETA5
	LACK	TR
	TBLR	TER
	LACK	GA
	TBLR	GAMA
	LACK	KA
	TBLR	KIN

* SEMICICLO POSITIVO

L1	LAC	RESUL
	SACL	DEV
	LAC	TETA1
	SACL	TET1
	LAC	TETA2
	SACL	TET2
	LAC	TETA3
	SACL	TETS

LAC TETA4
 SACL TET4
 LAC TETA5
 SACL TETS
 LT BETA
 LACK DAT1
 TBLR ENT
 CALL SUB3
 LAC CDAT
 SUB DELTA
 AND MASK3
 SACL CDAT
 BNZ L1
 LACK CT
 TBLR CDAT

* BEMICICLO NEGATIVO

L2 LAC RESUL
 SACL DEV
 LAC TETA1
 SACL TET1
 LAC TETA2
 SACL TET2
 LAC TETA3
 SACL TET3
 LAC TETA4
 SACL TET4
 LAC TETA5
 SACL TETS
 LT BETA
 LACK DAT2
 TBLR ENT
 CALL SUB3
 LAC CDAT
 SUB DELTA
 AND MASK3
 SACL CDAT
 BNZ L2
 LACK CT
 TBLR CDAT
 B L1

* SUBRUTINA DE CALCULO

**SE OBTIENE Wm

SUB3 MPY ENT
 PAC
 SACH SUM2,4
 LT ALPHA
 MPY DEV
 PAC
 SACH SUM1,4
 LAC SUM1
 ADD SUM2
 SACL RESUL

**SE OBTIENE DERIVADA DE Wm

SUB	DEV
SACL	MULT
LT	MULT
MPY	INV
PAC	
SACL	DER

**SE LEE MUESTRA DE W

ALOOP	BIOZ	ALOOP1
	B	ALDOP
ALOOP1	IN	VEL,2

**SE OBTIENE PRODUCTO (PERIODO) (GAMA) (W - Wm)

LAC	VEL
SUB	RESUL
SACL	ERR
LT	ERR
MPY	TER
PAC	
SACH	TEMP,4
LT	TEMP
MPY	GAMA
PAC	
SACH	DIF,4

**SE OBTIENE PRODUCTO (DIF) (DER Wm)

LT	DIF
MPY	DER
PAC	
SACH	DIF1,4

**SE OBTIENE PRODUCTO (DIF) (Wm)

LT	DIF
MPY	RESUL
PAC	
SACH	DIF2,4

LAC	VEL
BGEZ	LB

**SI W MENOR QUE CERO

LACK	0	
SACL	I1	* I1=0
LACK	IPDS	
TBLR	I2	* I2=1
LAC	TET2	
SACL	TETA2	* SE OBTIENE TETA2
LAC	TET3	
SUB	DIF2	
SACL	TETA3	* SE OBTIENE TETA3

LAC TET4
SACL TETA4 * SE OBTIENE TETA4
LAC TETS
SUB DIF
SACL TETAS * SE OBTIENE TETAS
B L7

**SI W MAYOR O IGUAL QUE CERO

8 LACK IPOS
TBLR I1 * I1=1
LACK 0
SACL I2 * I2=0
LAC TET2
SUB DIF2
SACL TETA2 * SE OBTIENE TETA2
LAC TET3
SACL TETA3 * SE OBTIENE TETA3
LAC TET4
SUB DIF
SACL TETA4 * SE OBTIENE TETA4
LAC TETS
SACL TETAS * SE OBTIENE TETAS

**SE OBTIENE T1=(TETA1) (DER Wm)

9 LAC TET1
SUB DIF1
SACL TETA1
LT DER
MPY TETA1
PAC
SACH TERM1,4

**SE OBTIENE T2=(I1) (TETA2) (Wm)

LT RESUL
MPY I1
PAC
SACH TEMP,4
LT TEMP
MPY TETA2
PAC
SACH TERM2,4

**SE OBTIENE T3=(I2) (TETA3) (Wm)

LT RESUL
MPY I2
PAC
SACH TEMP,4
LT TEMP
MPY TETA3
PAC
SACH TERM3,4

**SE OBTIENE T4=(I1) (TETA4)

LT I1
 MPY TETA4
 PAC
 SACH TERM4,4

*SE OBTIENE T5=(I2) (TETA5)

LT I2
 MPY TETA5
 PAC
 SACH TERMS,4

*SE OBTIENE I=1/K (T1+T2+T3+T4+T5)

LAC TERM1
 ADD TERM2
 ADD TERM3
 ADD TERM4
 ADD TERMS
 SACL CORR
 LT KIN
 MPY CORR
 PAC
 SACH CUAD,4
 OUT CUAD,2
 CALL SUB
 CALL SUB
 RET

SUBROUTINA DE RETARDO

SUB B START
 CONT DATA 2
 CONTA EQU 41
 START LDPK 0
 LACK CONT
 TBLR CONTA
 4 CALL SUB1
 LAC CONTA
 SUB DELTA
 AND MASK3
 SACL CONTA
 BZ L5
 B L4
 5 RET
 SUB1 B INIC
 CON DATA >7FFF
 CONTD EQU 42
 INIC LDPK 0
 LACK CON
 TBLR CONTD
 7 LAC CONTD
 SUB DELTA
 AND MASK3
 SACL CONTD
 BZ L6
 B L7

RET
END

71

* PROGRAMA CONTROL ADAPTABLE CON CINCO PARAMETROS
* (Wf SENAL SENOIDAL)

* DATOS PARA LA GENERACION DE UNA ONDA SENOIDAL

SINE	B	PRIN
	DATA	>0
	DATA	>324
	DATA	>646
	DATA	>964
	DATA	>C7C
	DATA	>F8D
	DATA	>1294
	DATA	>1590
	DATA	>187E
	DATA	>1B5D
	DATA	>1E2B
	DATA	>20E7
	DATA	>238E
	DATA	>2620
	DATA	>2B9A
	DATA	>2AFB
	DATA	>2D41
	DATA	>2F6C
	DATA	>3179
	DATA	>3368
	DATA	>3537
	DATA	>36E5
	DATA	>3871
	DATA	>39DB
	DATA	>3B21
	DATA	>3C42
	DATA	>3D3F
	DATA	>3E15
	DATA	>3ECS
	DATA	>3F4F
	DATA	>3FB1
	DATA	>3FEC
	DATA	>4000
	DATA	>3FEC
	DATA	>3FB1
	DATA	>3F4F
	DATA	>3ECS
	DATA	>3E15
	DATA	>3D3F
	DATA	>3C42
	DATA	>3B21
	DATA	>39DB
	DATA	>3871
	DATA	>36E5
	DATA	>3537
	DATA	>3368
	DATA	>3179
	DATA	>2F6C
	DATA	>2D41
	DATA	>2AFB
	DATA	>2B9A

DATA >2620
DATA >23BE
DATA >20E7
DATA >1E2B
DATA >1B5D
DATA >187E
DATA >1590
DATA >1294
DATA >FBD
DATA >C7C
DATA >964
DATA >646
DATA >324
DATA >0
DATA >FCDC
DATA >F9BA
DATA >F69C
DATA >F3B4
DATA >F073
DATA >ED6C
DATA >EA70
DATA >E7B2
DATA >E4A3
DATA >E1D5
DATA >DF19
DATA >DC72
DATA >D9E0
DATA >D766
DATA >D505
DATA >D2BF
DATA >D094
DATA >CEB7
DATA >CC9B
DATA >CAC9
DATA >C91B
DATA >C7BF
DATA >C625
DATA >C4DF
DATA >C3BE
DATA >C2C1
DATA >C1EB
DATA >C13B
DATA >C0B1
DATA >C04F
DATA >C014
DATA >C000
DATA >C014
DATA >C04F
DATA >C0B1
DATA >C13B
DATA >C1EB
DATA >C2C1
DATA >C3BE
DATA >C4DF
DATA >C625
DATA >C7BF
DATA >C91B

DATA >CAC9
 DATA >CC98
 DATA >CEB7
 DATA >D094
 DATA >D2BF
 DATA >D505
 DATA >D766
 DATA >D9E0
 DATA >DC72
 DATA >DF19
 DATA >E1D5
 DATA >E4A3
 DATA >E782
 DATA >EA70
 DATA >ED6C
 DATA >F073
 DATA >F384
 DATA >F69C
 DATA >F9BA
 DATA >FCDC

M1 DATA >7FFF
 DEL DATA 1
 CONT DATA 1
 CON DATA >3FF
 C1 DATA 4
 M4 DATA >7FFF
 KA DATA 4096
 GA DATA 4096
 TR DATA 26
 ALPH DATA 4069
 BET DATA 26
 TINV DATA 156
 IPOS DATA 4096
 RATE DATA 499
 TE1 DATA 0
 TE2 DATA 0
 TE3 DATA 0
 TE4 DATA 0
 TES DATA 0

* LOCALIDADES UTILIZADAS EN LA MEMORIA DE DATOS

DELTA EQU 0
 ALFA EQU 1
 SINA EQU 2
 MASK EQU 3
 OFFSET EQU 4
 GAMA EQU 5
 TER EQU 6
 ERR EQU 7
 TEMP EQU 8
 DIF EQU 9
 TETA1 EQU 10
 TETA2 EQU 11
 TETA3 EQU 12
 TETA4 EQU 13

TETA5	EQU	14
TET1	EQU	15
TET2	EQU	16
TET3	EQU	17
TET4	EQU	18
TET5	EQU	19
ALPHA	EQU	20
BETA	EQU	21
SUM1	EQU	22
SUM2	EQU	23
ENT	EQU	24
RESUL	EQU	25
DEV	EQU	26
INV	EQU	27
MULT	EQU	28
DER	EQU	29
TERM1	EQU	30
TERM2	EQU	31
TERM3	EQU	32
TERM4	EQU	33
TERM5	EQU	34
I1	EQU	35
I2	EQU	36
CORR	EQU	37
VEL	EQU	38
KIN	EQU	39
SEN	EQU	40
CONT1	EQU	41
CONTA	EQU	42
CONTD	EQU	43
MASK4	EQU	44
DIF1	EQU	45
DIF2	EQU	46
PRIN	LDPK	0
	LACK	250
	SACL	0
	OUT	0,0
	LACK	RATE
	TBLR	0
	OUT	0,1
	LACK	ALPH
	TBLR	ALPHA
	LACK	BET
	TBLR	BETA
	LACK	TINV
	TBLR	INV
	LACK	DEL
	TBLR	DELTA
	LACK	M1
	TBLR	MASK
	LACK	SINE
	SACL	OFFSET
	ZAC	
	SACL	RESUL
	SACL	ALFA
	LACK	TE1
	TBLR	TETA1

LACK TE2
 TBLR TETA2
 LACK TE3
 TBLR TETA3
 LACK TE4
 TBLR TETA4
 LACK TE5
 TBLR TETA5
 LACK TR
 TBLR TR
 LACK GA
 TBLR GAMA
 LACK KA
 TBLR KIN
 LACK C1
 TBLR CUNT1

* PROGRAMA PRINCIPAL

L1 LACK C1
 TBLR CONT1
 L2 LAC RESUL
 SACL DEV
 LAC TETA1
 SACL TET1
 LAC TETA2
 SACL TET2
 LAC TETA3
 SACL TET3
 LAC TETA4
 SACL TET4
 LAC TETA5
 SACL TET5
 CALL SWAVE1
 LAC CONT1
 SUB DELTA
 AND MASK4
 SACL CONT1
 BNZ L2
 LAC SINA
 SACL ENT
 LT BETA
 CALL SUB3
 B L1

* SUBROUTINA QUE GENERA WF

SWAVE1 LAC ALFA,8
 SACH TEMP
 LAC TEMP
 ADD OFFSET
 TBLR SINA
 LAC ALFA
 ADD DELTA
 AND MASK
 SACL ALFA
 CALL SUB

RET

* SUBROUTINA DE CALCULO

**SE OBTIENE Wm

SUB3	MPY	ENT
	PAC	
	SACH	SUM2,4
	LT	ALPHA
	MPY	DEV
	PAC	
	SACH	SUM1,4
	LAC	SUM1
	ADD	SUM2
	SACL	RESUL

**SE OBTIENE DERIVADA DE Wm

	SUB	DEV
	SACL	MULT
	LT	MULT
	MPY	INV
	PAC	
	SACL	DER

**SE LEE MUESTRA DE W

ALOOP	BIOZ	ALOOP1
	B	ALOOP
ALOOP1	IN	VEL,2

**SE OBTIENE PRODUCTO (PERIODO) (GAMA) (W - Wm)

LAC	VEL
SUB	RESUL
SACL	ERR
LT	ERR
MPY	TER
PAC	
SACH	TEMP,4
LT	TEMP
MPY	GAMA
PAC	
SACH	DIF,4

**SE OBTIENE PRODUCTO (DIF) (DER Wm)

LT	DIF
MPY	DER
PAC	
SACH	DIF1,4

**SE OBTIENE PRODUCTO (DIF) (Wm)

LT	DIF
MPY	RESUL
PAC	

SACH DIF2,4
LAC VEL
BGEZ L8

**SI W MENOR QUE CERO

LACK 0
SACL I1 * I1=0
LACK IPOS
TBLR I2 * I2=1
LAC TET2
SACL TETA2 * SE OBTIENE TETA2
LAC TET3
SUB DIF2
SACL TETA3 * SE OBTIENE TETA3
LAC TET4
SACL TETA4 * SE OBTIENE TETA4
LAC TET5
SUB DIF
SACL TETA5 * SE OBTIENE TETA5
B L9

**SI W MAYOR O IGUAL QUE CERO

B LACK IPOS
TBLR I1 * I1=1
LACK 0
SACL I2 * I2=0
LAC TET2
SUB DIF2
SACL TETA2 * SE OBTIENE TETA2
LAC TET3
SACL TETA3 * SE OBTIENE TETA3
LAC TET4
SUB DIF
SACL TETA4 * SE OBTIENE TETA4
LAC TET5
SACL TETA5 * SE OBTIENE TETA5

*SE OBTIENE T1=(TETA1) (DER Wm)

9 LAC TET1
SUB DIF1
SACL TETA1
LT DER
MPY TETA1
PAC
SACH TERM1,4

*SE OBTIENE T2=(I1) (TETA2) (Wm)

LT RESULT
MPY I1
PAC
SACH TEMP,4
LT TEMP
MPY TETA2

FAC
SACH TERM2,4

**SE OBTIENE T3=(I2)(TETA3)(Wm)

LT RESUL
MPY I2
FAC
SACH TEMP,4
LT TEMP
MPY TETA3
FAC
SACH TERM3,4

**SE OBTIENE T4=(I1)(TETA4)

LT I1
MPY TETA4
FAC
SACH TERM4,4

**SE OBTIENE T5=(I2)(TETA5)

LT I2
MPY TETA5
FAC
SACH TERMS,4

**SE OBTIENE I=1/K (T1+T2+T3+T4+T5)

LAC TERM1
ADD TERM2
ADD TERM3
ADD TERM4
ADD TERMS
SACL CORR
LT KIN
MPY CORR
FAC
SACH SEN,4
OUT SEN,2
RET

* SUBROUTINA DE RETARDO

SUB LDPK 0
 LACK CONT
 TBLR CONTA
 LACK M4
L4 TBLR MASK4
 CALL SUB1
 LAC CONTA
 SUB DELTA
 AND MASK4
 SACL CONTA
 BZ L5
 B L4

5
JB1
7
6
RET
LDPK
LACK
TBLR
LAC
SUB
AND
SACL
BZ
B
RET
END

O
CON
CONTD
CONTD
DELTA
MASK4
CONTD
L6
L7

PROGRAMA PARA GRAFICAR PARAMETROS

```

B      PRIN
M3     DATA  :7FFF
C3     DATA  7139
CONTS  EQU    0
MASK3  EQU    1
DELTA  EQU    2
PARAM  EQU    3
PARAM1 EQU    4
PRIN   LDPK   0

```

* CARGA REGISTRO DE CONTROL

```

LACK   SB
SACL   0
OUT    0,0

```

* CARGA DIRECCION DE INICIO DE LA MEMORIA DE EXPANSION

```

LACK   0
SACL   0
OUT    0,4

```

```

LACK   C3
TBLR   CONS
LACK   M3
TBLR   MASK3
LACK   1
SACL   DELTA

```

* LEE DATO DE LA MEMORIA DE EXPANSION

```

L1     IN     PARAM,5
      LAC    PARAM
      SACL   PARAM1

```

* SALIDA DE DATO POR EL PUERTO 2

```

OUT    PARAM1,2
CALL   SUB
CALL   SUB

```

```

LAC    CONS
SUB    DELTA
AND    MASK3
SACL   CONS
BNZ    L1
B      L2

```

* SUBROUTINA DE RETARDO

```

SUB     B      START
CON     DATA  1
CONTA   EQU    5
START   LDPK   0

```

L4	LACK	CON
	TBLR	CONTA
	CALL	SUB1
	LAC	CONTA
	SUB	DELTA
	AND	MASK3
	SACL	CONTA
	BZ	L5
	B	L4
L5	RET	
SUB1	B	INIC
CON	DATA	>FFF
CONTD	EQU	6
INIC	LDPK	0
	LACK	CON
	TBLR	CONTD
L7	LAC	CONTD
	SUB	DELTA
	AND	MASK3
	SACL	CONTD
	BZ	L6
	B	L7
L6	RET	
L2	ZAC	
	END	

PROGRAMA 5

PROGRAMA UTILIZADO PARA GRAFICAR UN NUMERO N DE DATOS

```

*
*
*
TETA  B   PRIN
      DATA 7087
      DATA 9144
      DATA 5221
      DATA 2842
      DATA 3107
      DATA 4354
      DATA 4777
      DATA 4738
      DATA 4616
      DATA 4672
M1    DATA >7FFF
BET   DATA 10
DEL   DATA 1
DELTA EQU 0
ALFA  EQU 1
TET   EQU 2
MASK  EQU 3
OFFSET EQU 4
BETA  EQU 5
PRIN  LDPK 0
      LACK 250
      SACL 0
      OUT  0,0
      LACK BET
      TBLR BETA
      LACK DEL
      TBLR DELTA
      LACK M1
      TBLR MASK
      LACK TETA
      SACL OFFSET
      ZAC
      SACL ALFA
L1    CALL TETA1
      OUT TET,2
      CALL SUB
      LAC  BETA
      SUB  DELTA
      AND  MASK
      SACL BETA
      BNZ  L1
      B    L2
TETA1 LAC  ALFA
      ADD  OFFSET
      TBLR TET
      LAC  ALFA
      ADD  DELTA
      SACL ALFA
      RET
SUB    B    START
CONT  DATA 6
CDNTA EQU 6
START LACK  CONT

```

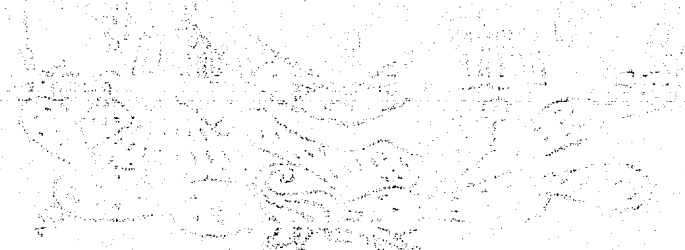
TBLR CONTA
CALL SUB1
LAC CONTA
SUB DELTA
AND MASK
SACL CONTA
SZ L5
S L4
RET
S INIC
DATA >7FFF.
EQU 7
LACK CON
TBLR CONTD
LAC CONTD
SUB DELTA
AND MASK
SACL CONTD
BZ L6
B L7
RET
ZAC
END

L4

L5
SUB1
CON
CONTD
INIC

L7

L6
L2



UNIVERSIDAD NACIONAL

****VARIABLES****

*** VARIABLES COMUNES A LOS PROGRAMAS**

GAMA	PERIODO DE MUESTREO T
TER	ERROR (W - Wm)
ERR	REGISTRO TEMPORAL
TEMP	(T) (GAMA) (W - Wm)
DIF	(DIF) (DER Wm)
DIF1	(DIF) (Wm)
DIF2	*PARAMETROS EN EL TIEMPO k
TETA1	*
TETA2	*
TETA3	*PARAMETROS EN EL TIEMPO k-1
TET1	*
TET2	*
TET3	$1/(1+aT)$
ALPHA	$aT/(1+aT)$
BETA	(ALPHA) (Wm(k-1))
SUM1	(BETA) (Wm(k))
SUM2	Wm(k)
ENT	VELOCIDAD DE REFERENCIA Wm(k)
RESUL	VELOCIDAD DE REFERENCIA Wm(k-1)
DEV	$1/T$
INV	DIFERENCIA Wm(k) - Wm(k-1)
MULT	DERIVADA DE Wm
DER	(TETA1) (DER Wm)
TERM1	(TETA2) (I1) (Wm)
TERM2	(TETA3) (I2) (Wm)
TERMS	
I1	
I2	VELOCIDAD W
VEL	CONSTANTE K
KIN	*CONTADORES PARA LA SUBR DE RETARDO
CONTA	*
CONTD	

*** VARIABLES QUE SE UTILIZAN SOLAMENTE EN EL PROG CONTROL ADAPTABLE
* CON TRES PARAMETROS (Wf SENAL CUADRADA)**

CDAT	CONTADOR
DELTA	MASCARA PARA CONTADOR CDAT
CORR	INCREMENTO PARA CONTADOR CDAT
CUAD	$I1+I2+I3$
	$I=1/K$ (CORR)

*** VARIABLES QUE SE UTILIZAN SOLAMENTE EN EL PROG CONTROL ADAPTABLE
* CON CINCO PARAMETROS (Wf SENAL CUADRADA)**

TETA4	*PARAMETROS EN EL TIEMPO k
TETA5	*
TET4	*PARAMETROS EN EL TIEMPO k-1
TET5	*
TERM4	(TETA4) (I1)
TERM5	(TETA5) (I2)
CORR	$T1+T2+T3+T4+T5$
CUAD	$I=1/K$ (CORR)

* VARIABLES QUE SE UTILIZAN SOLAMENTE EN LA SUBR DE GENERACION DE W_r
* EN EL PROG CONTROL ADAPTABLE CON CINCO PARAMETROS (SENAL SENCIDAL)

ALFA	CONTADOR
DELTA	INCREMENTO PARA CONTADOR ALFA
MASK	MASCARA PARA CONTADOR ALFA
OFSET	LOCALIDAD EN LA MP DEL PRIMER DATO
SINA	DATO DE LA SENCIDAL

* VARIABLES QUE SE UTILIZAN SOLAMENTE EN LA SUBR DE CALCULO EN EL PROG
* CONTROL ADAPTABLE CON CINCO PARAMETROS (SENAL SENCIDAL)

CONT1	CONTADOR PARA EL MUESTREO DE W _r
MASK4	MASCARA PARA LOS CONTADORES CONTA CONTD Y CONT1
TETA4	*PARAMETROS EN EL TIEMPO K
TETA5	*
TET4	*PARAMETROS EN EL TIEMPO k-1
TET5	*
TERM4	(TETA4) (I1)
TERM5	(TETA5) (I2)
CORR	T1+T2+T3+T4+T5
SEN	I=1/K (CORR)

APENDICE C
NOTAS TECNICAS DEL TMS32010

UNIVERSIDAD NACIONAL

Precision Digital Sine-Wave Generation with the TMS32010

INTRODUCTION

Sine-wave generators are fundamental building blocks of signal processing systems which are used in diverse applications, such as communication, instrumentation, and control. In the past, engineers usually designed these oscillators with analog circuitry. Now, however, new high-speed digital signal processors like the TMS32010 present designers with an alternative that in many cases is superior. The TMS32010 provides the speed and accuracy to produce stable, low-distortion sine waves over a wide range of frequencies.

This application report describes two different methods for implementing a digital sine wave generator using the TMS32010. The first method is a fast direct table lookup scheme suitable for applications not requiring extreme accuracy. The second approach, an enhancement of the first, includes linear interpolation to provide sine waveforms with a minimum of harmonic distortion.

DIRECT TABLE LOOKUP METHOD

The first algorithm is a simple, fast table lookup scheme. The sine values for N angles which are uniformly spaced around the unit circle are stored in a table which has the following format:

INDEX	ANGLE	SINE TABLE
0	$0 \times 360^\circ/N$	$S(0) = \sin(0^\circ/N)$
1	$1 \times 360^\circ/N$	$S(1) = \sin(360^\circ/N)$
2	$2 \times 360^\circ/N$	$S(2) = \sin(720^\circ/N)$
.	.	.
.	.	.
$N-2$	$(N-2) \times 360^\circ/N$	$S(N-2) = \sin((N-2) \times 360^\circ/N)$
$N-1$	$(N-1) \times 360^\circ/N$	$S(N-1) = \sin((N-1) \times 360^\circ/N)$

A sine wave is generated by stepping through the table at a constant rate (in effect, moving counterclockwise around the unit circle), wrapping around at the end of the table whenever 360° is exceeded. Using the table index as the angle parameter and DELTA as the step size, this lookup method generates the sequence:

$$S[\text{mod}(k \times \text{DELTA}, N)] \quad \text{for } k = 1, 2, 3, 4, \dots$$

where $\text{mod}(a,b) =$ remainder of the division a/b when this quotient is computed as an integer (e.g., $\text{mod}(22.54, 5) = 2.34$)

The 'mod' operator provides the wraparound at the end of the table. Figure 1 illustrates this algorithm.

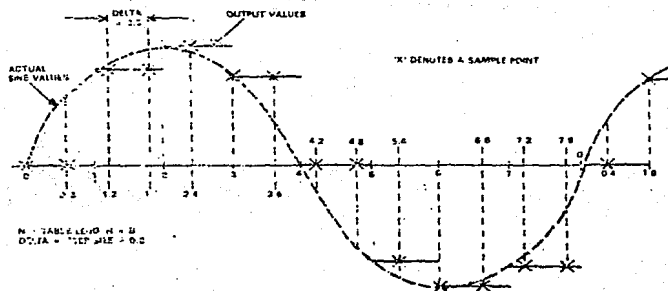


Figure 1. Direct Table Lookup

The sampled waveform generated is only an approximation to a sampled sinusoid. In general, the longer the table is the more resolution it provides, and consequently, the closer the approximation will be.

The frequency, f , of the sine wave depends on two factors:

- (1) The time interval between successive samples, i.e., the sampling interval, t
- (2) The step size, DELTA

f is given by the equation:

$$f = \frac{\text{DELTA}}{t \times N} \text{ [Hz]} \quad \text{where } t \text{ is expressed in seconds}$$

Note that to satisfy the Nyquist criterion there must be at least two samples generated each sinusoid period. This requires that $\text{DELTA} \leq N/2$.

In Figure 1, $N = 8$ and $\text{DELTA} = 0.6$. If, for instance, eight samples are generated each millisecond, then $t = 0.000125$ seconds and

$$f = \frac{0.6}{8 \times 0.000125} \text{ Hz} = 600 \text{ Hz}$$

TMS32010 Implementation

This section describes the concise TMS32010 subroutine, given in Appendix B, which implements the table lookup scheme based on a sine table with 128 entries. Each time this subroutine is called, the next sample point is calculated. This subroutine uses:

- (1) 138 (= 128 + 10) words of program memory space (128 words for sine table storage and 10 words for program memory)
- (2) 6 words in data memory as working registers

If this program is used as a subroutine, each sample can be computed in 3.0 microseconds. However, if the code is inserted directly in line with the code of a master program, avoiding the overhead of a subroutine, a sample can be computed in 2.2 microseconds:

The values in the sine table are all scaled. The decimal values, +1.0 and -1.0, are represented by the two's complement hexadecimal values 4000 and C000, respectively. All other values are scaled and rounded to the closest hexadecimal number. Rounding is used, rather than truncation, to avoid adding unnecessary distortion.

The 16-bit data memory location 'ALPHA' serves as a modulo 128 counter, which cycles through the sine table to select the sample points. ALPHA is regarded as having an integer and fractional part with the format:

Q Q Q Q Q Q Q Q . Q Q Q Q Q Q Q Q
15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0

The 16-bit data memory location 'DELTA' contains the step size. DELTA has the same (integer, fraction) format as ALPHA. Every time the sine wave subroutine is called, the contents of ALPHA are incremented by the contents of DELTA. The integer portion of ALPHA (i.e., the eight MSBs) is the pointer to the sine table. However, because the table starts at address location SINE, this pointer is offset by the value for that address before the table is accessed. The eight most significant bits of ALPHA are masked when ALPHA is updated to insure that they never exceed 127. The routine returns the sine value in the data memory location 'SINA'.

For any given sampling interval, t , the frequencies which can be generated must be of the form

$$f = \frac{\text{DELTA}}{t \times 128} \text{ [Hz]} \quad \text{where } t \text{ is expressed in seconds}$$

Since DELTA has a precision of eight bits to the right of the decimal place, any desired frequency ($\approx 1/2t$ [Hz]) can be approximated with an error of no more than

$$\frac{1/256}{t \times 128} \text{ [Hz]} = \frac{1}{32768 \times t} \text{ [Hz]}$$

For example, if the sampling frequency is 8 kHz, then the frequency resolution is

$$\frac{8000}{32768} \text{ Hz} = 0.25 \text{ Hz}$$

Harmonic Distortion

Due to approximations made in calculating the samples of a sine wave of frequency f , a certain amount of the "energy" of the samples' waveform will fall into other frequencies as well. These frequencies are either:

- (1) Harmonic frequencies, nf , where $n = 2, 3, 4, \dots$, or
- (2) Subharmonic frequencies, nf/m , where n and m are integers.

This spurious energy results in noise which is referred to as "harmonic distortion." It is usually measured in terms of Total Harmonic Distortion (THD) which is defined as the ratio

$$\text{THD} = \frac{\text{spurious harmonic energy}}{\text{total energy of the waveform}}$$

There are two sources of error in the table lookup algorithm which cause harmonic distortion:

- (1) Quantization error is introduced by representing the sine table values by 16-bit numbers.
- (2) Larger errors are introduced when points between table entries are sampled. This occurs when DELTA is not an integer.

The longer the sine table is, the less significant the second error source will be. Consequently, harmonic distortion decreases with increasing table length. Furthermore, when DELTA is an integer, quantization is the only error source, and THD is extremely small regardless of table size. THD is given for several table lengths and values of DELTA in Figure 2. Note that the figures in this table only represent the THD in the digitized sine wave, if the sine wave is reconstructed using a digital-to-analog converter and analog filters, these analog devices will contribute additional distortion. (The procedure for computing THD is described in Appendix A.)

LINEAR INTERPOLATION METHOD

To decrease the harmonic distortion for a given table size, an interpolation scheme can be used to compute the sine values between table entries more accurately. Linear interpolation is the simplest method to implement. This method uses the values of two consecutive table entries as the end points of a line segment. Sample points for parameter values falling between table entries assume values on the line segment between the points. This algorithm is illustrated in Figure 3.

TABLE LENGTH: 32

DELTA	THD
2.0	0.00000024
2.25	0.00300893
2.50	0.00240751
2.75	0.00300917
3.0	0.00000024
8.25	0.00300924
11.625	0.00315807

TABLE LENGTH: 64

DELTA	THD
2.0	0.00000048
2.25	0.00075269
2.50	0.00050219
2.75	0.00075233
3.00	0.00000018
8.25	0.00075204
11.625	0.00079078

TABLE LENGTH: 128

DELTA	THD
2.00	0.00000054
2.25	0.00018859
2.50	0.00015080
2.75	0.00018835
3.00	0.00000012
8.25	0.00018889
11.625	0.00020128

Figure 2. Total Harmonic Distortion Using Direct Table Lookup

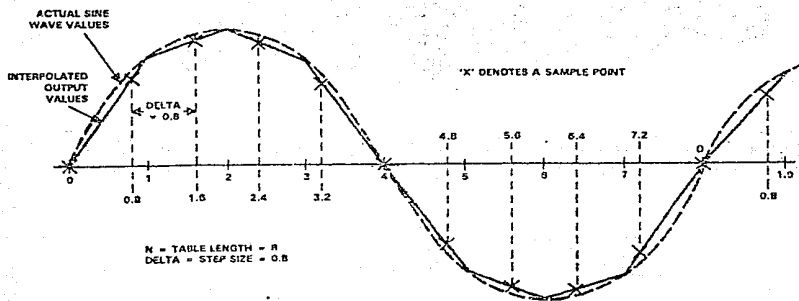


Figure 3. Linear Interpolation